BAB 3
PELAKSANAAN KERJA MAGANG

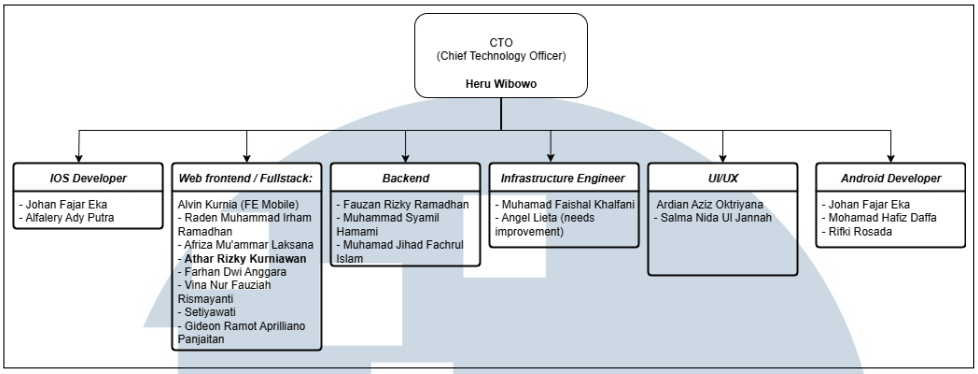
3.1 Kedudukan dan Koordinasi

Selama menjalani kerja magang di PT Svara Inovasi Indonesia, penulis ditempatkan pada divisi *Technology* di bawah koordinasi langsung Chief Technology Officer (CTO). Dalam divisi ini, penulis berperan sebagai *Frontend Engineer Intern* yang bertanggung jawab dalam perancangan serta implementasi antarmuka pengguna berbasis web untuk sistem RRI AI. Sistem RRI AI merupakan salah satu inisiatif strategis perusahaan dalam mendukung transformasi digital Radio Republik Indonesia melalui pemanfaatan teknologi kecerdasan buatan.

Pelaksanaan pengembangan sistem dilakukan dengan pendekatan kerja iteratif yang menyerupai metode *Agile Development*. Setiap fitur dikembangkan secara bertahap melalui siklus perancangan, implementasi, evaluasi, dan perbaikan berdasarkan masukan dari tim terkait. Dalam konteks ini, koordinasi lintas tim menjadi faktor krusial agar pengembangan frontend tetap selaras dengan kebutuhan backend dan layanan AI.

Dalam menjalankan tugasnya, penulis berkoordinasi secara langsung dengan beberapa tim, yaitu tim UI/UX Designer yang bertanggung jawab atas perancangan visual dan pengalaman pengguna, tim Backend Python/AI yang mengembangkan layanan pemrosesan audio, NLP, serta sistem rekomendasi berbasis Pydantic AI, serta tim Data Curation yang mengelola struktur dan validitas konten audio. Sinergi antar tim ini diperlukan agar antarmuka yang dikembangkan tidak hanya menarik secara visual, tetapi juga fungsional dan terintegrasi dengan baik.

Kegiatan magang dilaksanakan dengan pola kerja *hybrid*. Pada tahap awal, penulis mengikuti kegiatan *Work From Office* (WFO) untuk orientasi dan sinkronisasi teknis. Selanjutnya, pengerjaan dilakukan melalui *Work From Home* (WFH) dengan komunikasi rutin melalui *Daily Stand Up* (DS) yang dilaksanakan setiap pukul 14.00 WIB. Melalui DS, penulis menyampaikan progres pekerjaan, kendala teknis, serta rencana kerja berikutnya.



Gambar 3.1. Struktur organisasi PT Svara Inovasi Indonesia dan posisi penulis

Gambar 3.1 menunjukkan bahwa penulis berada pada jalur pengembangan frontend yang berkoordinasi langsung dengan CTO serta tim backend dan AI. Struktur ini memperjelas alur komunikasi, pembagian tanggung jawab, serta mekanisme eskalasi teknis selama pelaksanaan magang.

3.1.1 Pendekatan Metodologi Pengembangan Agile

Pengembangan sistem RRI AI Web menerapkan pendekatan *Agile Development* dengan pola kerja menyerupai metode *Scrum-like Agile*. Pendekatan ini dipilih karena kebutuhan sistem yang dinamis, khususnya pada pengembangan antarmuka pengguna dan integrasi layanan kecerdasan buatan yang memerlukan penyesuaian berkelanjutan berdasarkan hasil evaluasi dan masukan tim.

Proses pengembangan dilakukan secara iteratif dalam siklus pendek yang mencakup tahap perencanaan, implementasi, evaluasi, dan perbaikan. Setiap iterasi diawali dengan penentuan fitur frontend yang akan dikembangkan, dilanjutkan dengan implementasi dan integrasi, kemudian dievaluasi bersama tim UI/UX dan backend untuk memastikan kesesuaian terhadap kebutuhan sistem.

Permasalahan komunikasi yang muncul akibat pola kerja *hybrid* dan *Work From Home* (WFH) diatasi melalui mekanisme *Daily Stand Up* (DS) yang dilaksanakan secara rutin. Melalui DS, setiap anggota tim menyampaikan progres pekerjaan, kendala yang dihadapi, serta rencana kerja berikutnya. Selain itu, komunikasi asinkron melalui *chat* internal digunakan untuk diskusi teknis dan klarifikasi kebutuhan secara cepat sehingga hambatan koordinasi dapat diminimalkan dan pengembangan sistem tetap berjalan efektif.

3.2 Tugas yang Dilakukan

Selama masa kerja magang, penulis bertanggung jawab dalam proses perancangan dan implementasi fitur frontend pada aplikasi RRI AI berbasis web. Adapun tugas yang dilakukan meliputi:

- Merancang antarmuka pengguna menggunakan Figma untuk halaman Home, AI Chat, Podcast Recommendations, Top Artist/Playlist Detail, dan Audio Player.
- Mengimplementasikan desain antarmuka ke dalam kode menggunakan Next.js, TypeScript, dan Tailwind CSS.
- Mengembangkan komponen UI modular seperti sidebar navigasi, bubble chat AI, kartu rekomendasi konten, daftar podcast dan playlist, serta komponen audio player.
- Melakukan integrasi frontend dengan API backend berbasis Python/AI, termasuk modul NLP dan Pydantic AI pada fitur AI Chat.
- Melakukan pengujian tampilan dan responsivitas antarmuka pada berbagai ukuran layar.
- Menyusun dokumentasi teknis frontend serta mencatat hasil evaluasi pada setiap iterasi pengembangan.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang dilakukan secara bertahap dan iteratif. Rincian kegiatan yang dilakukan setiap minggu disajikan pada Tabel 3.1.

Tabel 3.1. Uraian pekerjaan yang dilakukan setiap minggu selama pelaksanaan kerja magang

Minggu Ke-	Uraian Pekerjaan
1	Orientasi lingkungan kerja, pengenalan struktur organisasi, serta pemahaman awal proyek RRI AI.
2	Analisis kebutuhan frontend dan pemahaman arsitektur sistem RRI AI.

Minggu Ke-	Uraian Pekerjaan
3	Penyusunan desain awal antarmuka Home dan AI Chat menggunakan Figma.
4	Pengembangan desain lanjutan untuk halaman konten dan diskusi revisi desain.
5	Setup proyek frontend dan implementasi layout dasar serta sidebar navigasi.
6	Pengembangan komponen UI modular dan penerapan styling menggunakan Tailwind CSS.
7	Implementasi halaman AI Chat termasuk bubble prompt dan respons AI.
8	Integrasi awal API backend untuk data radio dan podcast.
9	Pengembangan halaman detail konten dan rekomendasi.
10	Implementasi dan pengujian komponen Audio Player.
11	Evaluasi UX dan perbaikan alur navigasi antar halaman.
12	Penyempurnaan responsivitas tampilan dan layout aplikasi.
13	Pengembangan lanjutan fitur AI Chat dengan rekomendasi audio.
14	Refactoring kode dan penyusunan dokumentasi frontend.
15	Pengujian fungsional aplikasi secara menyeluruh.
16	Evaluasi proses pengembangan dan penyusunan laporan kerja magang.

Tabel 3.1 menguraikan rincian pekerjaan yang dilakukan oleh penulis selama menjalani masa magang di PT Svara Inovasi Indonesia pada proyek RRI AI. Tabel tersebut menyajikan pembagian aktivitas mingguan mulai dari tahap orientasi, analisis kebutuhan, perancangan desain UI/UX, implementasi frontend, integrasi API, hingga tahap pengujian dan penyusunan dokumentasi. Secara umum, pekerjaan penulis berfokus pada pengembangan antarmuka web responsif menggunakan *Next.js*, *TypeScript*, dan *Tailwind CSS*, serta integrasi dengan layanan backend Python/AI untuk fitur AI Chat dan rekomendasi konten audio.

3.3.1 Minggu 1 hingga Minggu 4: Orientasi, Analisis, dan Perancangan UI/UX

Pada minggu-minggu awal, penulis berfokus pada adaptasi lingkungan kerja dan pemahaman konteks proyek. Setelah orientasi, penulis mulai menganalisis

kebutuhan sistem dan menyusun rancangan awal antarmuka menggunakan Figma sebagai dasar visual sebelum masuk ke tahap implementasi.

- Minggu 1: Penulis mengikuti orientasi kerja, memahami struktur organisasi, alur komunikasi tim, serta pengenalan awal terhadap proyek RRI AI.
- Minggu 2: Penulis melakukan analisis kebutuhan frontend dan mempelajari arsitektur sistem RRI AI untuk memahami alur data serta integrasi dengan backend.
- Minggu 3: Penulis menyusun desain awal antarmuka halaman Home dan AI Chat menggunakan Figma dengan fokus pada layout utama dan komponen inti.
- Minggu 4: Penulis melanjutkan desain untuk halaman konten lain serta melakukan diskusi revisi desain dengan tim UI/UX untuk memastikan konsistensi tampilan dan kebutuhan fitur.

3.3.2 Minggu 5 hingga Minggu 8: Setup Frontend, Pengembangan Komponen, dan Integrasi Awal API

Pada fase ini, penulis mulai membangun fondasi proyek frontend serta mengembangkan komponen UI yang modular. Setelah struktur aplikasi terbentuk, penulis mengimplementasikan halaman AI Chat dan melakukan integrasi awal dengan backend untuk kebutuhan data radio dan podcast.

- Minggu 5: Penulis melakukan setup proyek frontend, menyusun struktur routing, serta mengimplementasikan layout dasar dan sidebar navigasi.
- Minggu 6: Penulis mengembangkan komponen UI secara modular dan menerapkan styling dengan Tailwind CSS agar tampilan konsisten dan mudah dikembangkan.
- Minggu 7: Penulis mengimplementasikan halaman AI Chat termasuk bubble prompt pada *new chat* dan struktur tampilan respons AI.
- Minggu 8: Penulis melakukan integrasi awal API backend untuk menampilkan data radio dan podcast, termasuk penanganan proses fetch data, *loading state*, dan pemetaan data ke komponen UI.

3.3.3 Minggu 9 hingga Minggu 12: Pengembangan Halaman Detail, Audio Player, dan Responsivitas

Pada minggu 9–12, penulis berfokus pada pengembangan halaman detail konten serta implementasi pemutar audio. Setelah fitur utama terbentuk, penulis melakukan evaluasi pengalaman pengguna dan menyempurnakan responsivitas tampilan agar nyaman digunakan pada berbagai ukuran layar.

- Minggu 9: Penulis mengembangkan halaman detail konten dan rekomendasi agar pengguna dapat membuka detail playlist/artist serta menelusuri daftar konten.
- Minggu 10: Penulis mengimplementasikan dan menguji komponen Audio Player untuk kebutuhan pemutaran radio/stream, termasuk kontrol dasar dan sinkronisasi state pemutaran.
- Minggu 11: Penulis melakukan evaluasi UX serta memperbaiki alur navigasi antar halaman berdasarkan hasil pengujian dan masukan tim.
- Minggu 12: Penulis menyempurnakan responsivitas tampilan dan layout aplikasi, terutama pada komponen kompleks seperti sidebar, panel rekomendasi, dan daftar konten.

3.3.4 Minggu 13 hingga Minggu 16: Penyempurnaan AI Chat, Refactoring, Pengujian, dan Laporan

Tahap akhir difokuskan pada penyempurnaan fitur AI Chat dan kualitas kode. Penulis melakukan refactoring untuk meningkatkan keterbacaan dan maintainability, menguji aplikasi secara menyeluruh, serta menyusun dokumentasi dan laporan magang sebagai hasil akhir kegiatan.

- Minggu 13: Penulis mengembangkan lanjutan fitur AI Chat agar respons AI dapat menampilkan rekomendasi audio secara lebih terstruktur dan mudah diakses.
- Minggu 14: Penulis melakukan refactoring kode, merapikan struktur komponen, serta mulai menyusun dokumentasi teknis frontend.
- Minggu 15: Penulis melakukan pengujian fungsional aplikasi secara menyeluruh untuk memastikan seluruh fitur berjalan stabil dan sesuai kebutuhan.

- Minggu 16: Penulis mengevaluasi proses pengembangan yang telah dilakukan dan menyusun laporan kerja magang sebagai dokumentasi akhir.

3.4 Perancangan Sistem

Sebelum memulai implementasi frontend, dilakukan perancangan sistem untuk memastikan struktur informasi dan alur interaksi pengguna sesuai dengan kebutuhan pengguna akhir. Tiga elemen penting yang dirancang adalah *Sitemap*, *Flowchart*, dan *Wireframe*. Perancangan dimulai dengan pembuatan *sitemap* yang menggambarkan struktur halaman dan relasi antar halaman dalam aplikasi RRI AI Web. Setelah *sitemap* selesai, dibuat *flowchart* yang menjelaskan alur navigasi pengguna dari satu halaman ke halaman lainnya. Selanjutnya, dilakukan perancangan *wireframe* sebagai visual awal dari masing-masing halaman berdasarkan layout komponen utama yang dirancang dalam *Figma*, kemudian hasil rancangan tersebut diimplementasikan ke dalam website.

3.4.1 Prinsip UI/UX pada Layanan Audio Berbasis Web

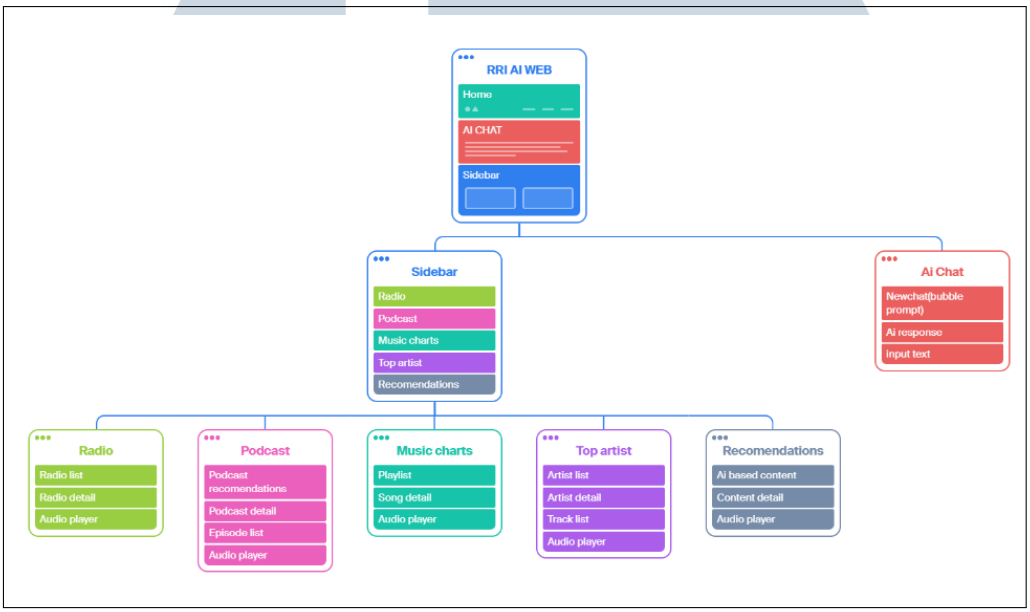
Perancangan antarmuka pada layanan audio berbasis web memiliki karakteristik yang berbeda dibandingkan dengan website informatif atau transaksional. Pada layanan audio, fokus utama pengguna tidak selalu berada pada layar, melainkan pada konten suara yang sedang diputar. Oleh karena itu, desain UI/UX harus mendukung pola penggunaan *audio-first experience*.

Beberapa prinsip utama yang diterapkan pada sistem RRI AI Web meliputi penyediaan pemutar audio global yang konsisten di seluruh halaman, pengurangan distraksi visual saat audio diputar, serta penyediaan kontrol pemutaran yang mudah dijangkau. Berbeda dengan website umum yang menuntut perhatian visual penuh, antarmuka audio web dirancang agar pengguna tetap dapat melakukan aktivitas lain tanpa kehilangan kendali terhadap audio.

Selain itu, hierarki visual dibuat sederhana dan intuitif agar pengguna dapat dengan cepat memahami status pemutaran, berpindah konten, atau kembali ke interaksi AI Chat. Pendekatan ini bertujuan meningkatkan kenyamanan, mengurangi beban kognitif, serta menjaga kesinambungan pengalaman pengguna pada layanan audio berbasis web.

A Perancangan Sitemap

Pada tahap ini, dibuat sebuah *sitemap* yang menggambarkan struktur halaman dan hubungan antar halaman dalam aplikasi RRI AI Web. Gambar berikut menunjukkan hasil perancangan struktur halaman, yang berfungsi sebagai acuan utama untuk navigasi dan pengelompokan fitur dalam sistem.



Gambar 3.2. *Sitemap* aplikasi RRI AI Web

Gambar 3.2 menunjukkan struktur halaman utama yang berpusat pada *Home/AI Chat* sebagai titik masuk pengguna. Dari halaman tersebut, pengguna dapat menavigasi ke fitur *Radio, Podcast, Music Charts, Top Artist*, serta *Recommendations* melalui *sidebar*. Masing-masing fitur memiliki halaman turunan seperti halaman daftar (*list*), halaman detail konten, hingga halaman pemutar audio (*audio player*) untuk memutar konten radio maupun audio lainnya.

A.1 Penjelasan Struktur Sitemap dan Alur Pengguna

Struktur sitemap aplikasi RRI AI Web dirancang dengan pendekatan *AI-first navigation*, di mana halaman *Home/AI Chat* berperan sebagai pusat interaksi utama pengguna. Pendekatan ini berbeda dengan website audio konvensional yang umumnya menempatkan halaman daftar konten sebagai titik awal.

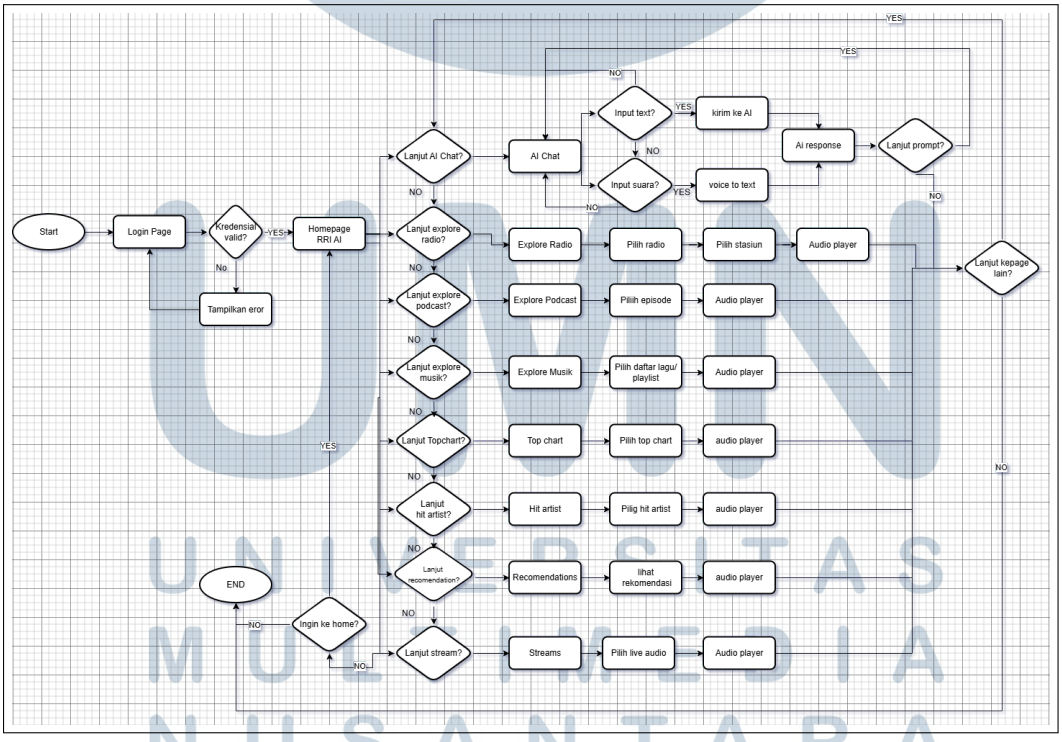
Dari halaman *Home/AI Chat*, pengguna dapat mengakses fitur utama seperti Radio, *Podcast Recommendations, Music Charts, Top Artist, dan Playlist* melalui

sidebar navigasi. Setiap fitur memiliki struktur hierarkis berupa halaman daftar (*list*) dan halaman detail konten. Halaman *Audio Player* berfungsi sebagai komponen lintas halaman yang dapat diakses dari berbagai konteks tanpa memutus pengalaman pengguna.

Struktur sitemap ini dirancang untuk meminimalkan perpindahan halaman yang tidak perlu serta menjaga kesinambungan antara interaksi berbasis AI dan eksplorasi konten audio. Dengan demikian, pengguna dapat berpindah dari percakapan AI menuju pemutaran audio secara intuitif dan efisien.

B Perancangan Flowchart

Flowchart berikut digunakan untuk menggambarkan alur interaksi pengguna pada aplikasi RRI AI Web. Flowchart ini memberikan visualisasi terhadap langkah-langkah utama pengguna ketika mengakses fitur AI Chat, menerima rekomendasi konten, serta berpindah menuju halaman detail atau halaman audio player.



Gambar 3.3. Flowchart navigasi dan interaksi pengguna pada aplikasi RRI AI Web

Gambar 3.3 menunjukkan bahwa alur utama dimulai dari halaman Home/AI Chat. Ketika pengguna membuat *new chat*, sistem menampilkan *bubble*

prompt awal sebagai bantuan interaksi. Selanjutnya pengguna dapat mengirim pertanyaan atau permintaan rekomendasi. Sistem AI memproses input tersebut dan menghasilkan respons, yang dapat berupa teks maupun rekomendasi konten audio. Dari rekomendasi tersebut, pengguna dapat mengakses halaman detail konten atau langsung memutar audio melalui komponen *audio player*. Alur ini dirancang agar interaksi AI dan eksplorasi konten tetap terhubung dalam satu pengalaman pengguna yang konsisten.

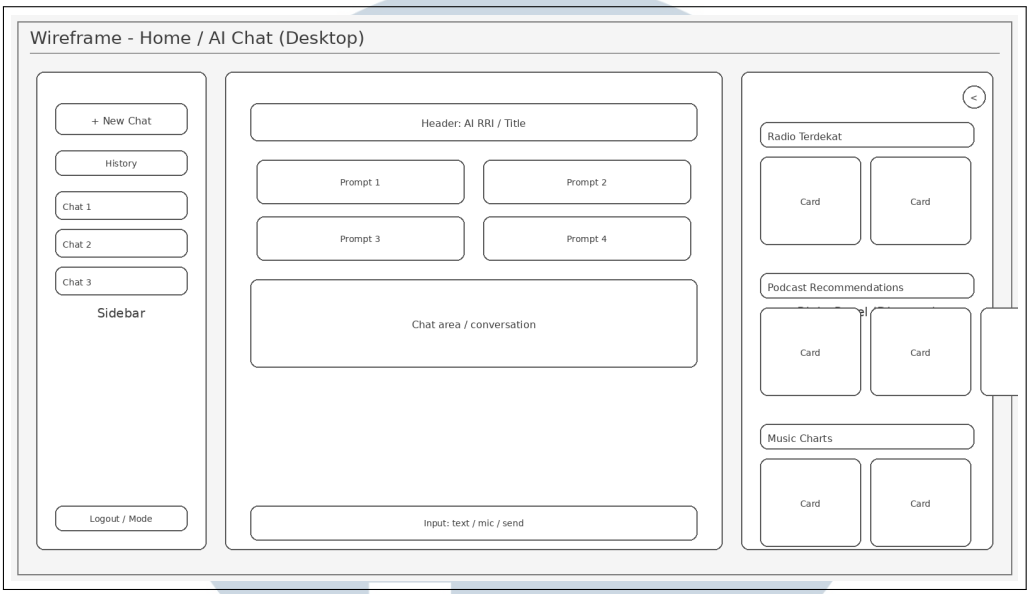
C Perancangan Wireframe

Wireframe adalah representasi visual awal dari tata letak (*layout*) halaman yang menunjukkan elemen-elemen utama dan struktur interaksi pengguna. Pada tahap ini, wireframe dibuat untuk memastikan posisi komponen inti seperti sidebar, area chat, kartu rekomendasi, daftar konten, serta pemutar audio sudah sesuai sebelum masuk ke desain detail (*high-fidelity*) di Figma dan implementasi ke website. Perancangan wireframe dibagi menjadi dua platform, yaitu versi web (desktop) dan versi mobile.

C.1 Perancangan Wireframe Versi Web (Desktop)

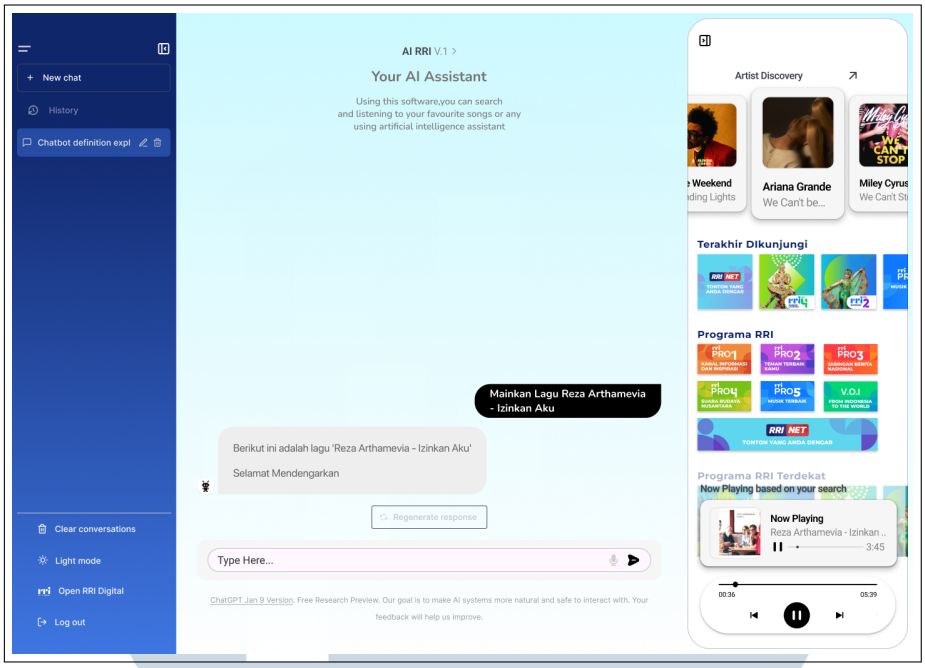
Pada versi web, rancangan antarmuka dibuat dengan pendekatan tiga kolom utama, yaitu sidebar navigasi di kiri, area utama (chat/konten) di tengah, dan panel discovery di kanan. Pola ini bertujuan agar pengguna dapat berinteraksi dengan AI sambil tetap mengeksplorasi rekomendasi konten tanpa berpindah halaman terlalu sering.

C.1.1 Halaman Home / AI Chat (Desktop)



Gambar 3.4. Wireframe (Low-Fidelity) halaman Home / AI Chat versi web (desktop)

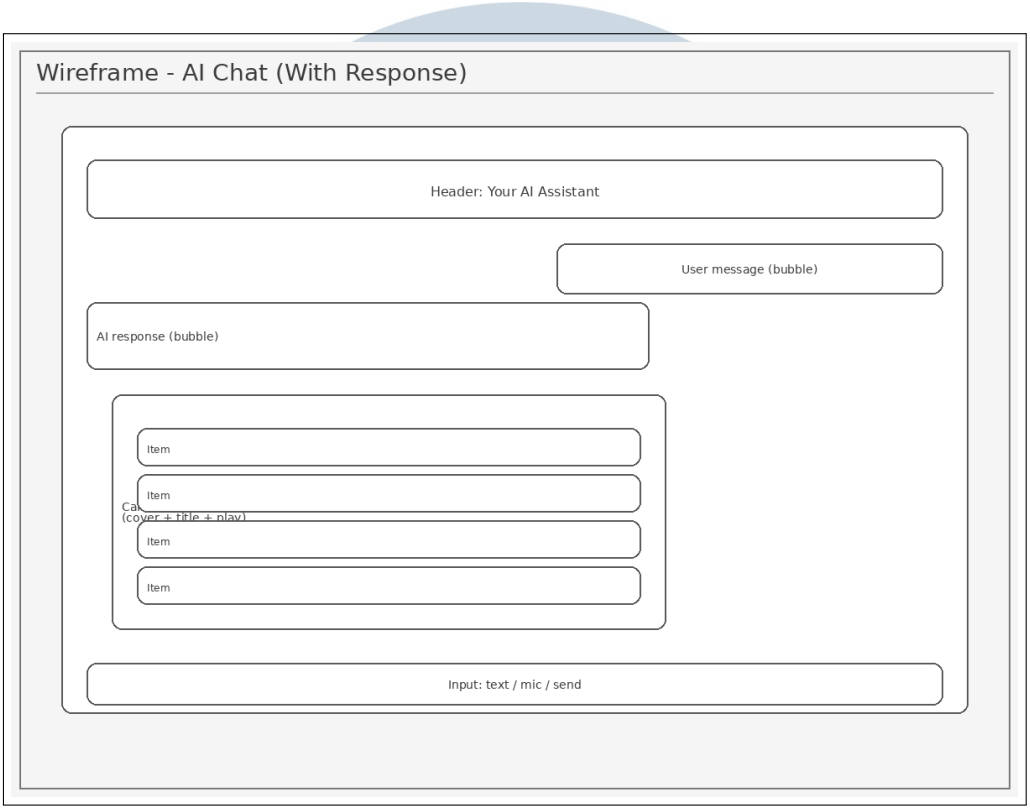
Gambar 3.4 menunjukkan struktur dasar halaman Home/AI Chat pada tampilan desktop. Area kiri digunakan sebagai sidebar untuk navigasi dan riwayat percakapan, area tengah sebagai ruang percakapan utama, sedangkan area kanan menampilkan panel discovery yang berisi rekomendasi konten (radio, podcast, music charts, dan artist).



Gambar 3.5. Desain (High-Fidelity) Figma halaman Home / AI Chat versi web

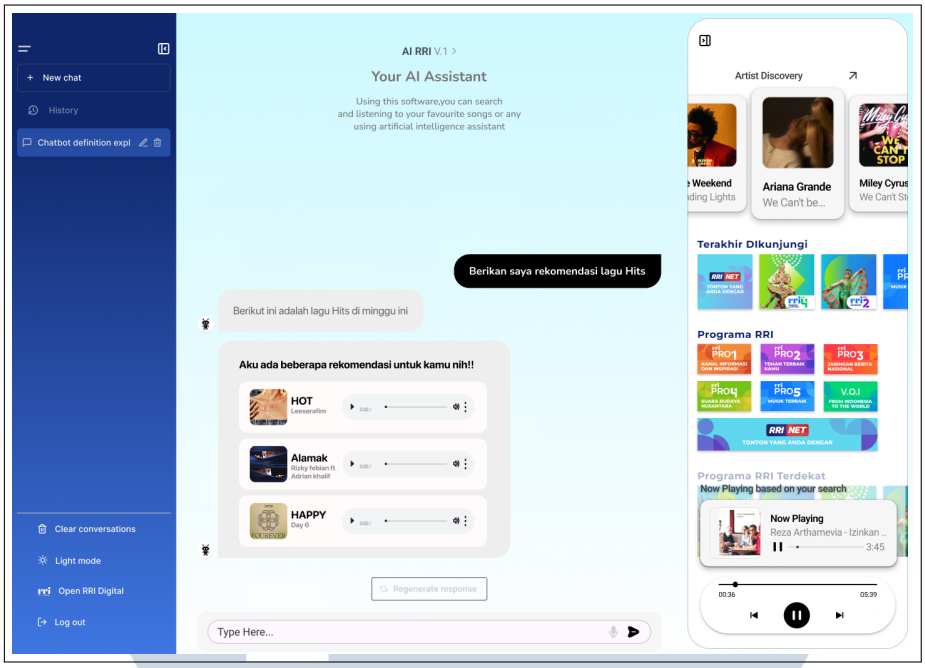
Gambar 3.5 merupakan hasil desain high-fidelity dari halaman Home/AI Chat pada Figma. Elemen visual seperti latar gradien, tipografi, jarak antar komponen, dan gaya tombol dibuat konsisten agar pengalaman pengguna terasa modern dan tetap mudah dibaca.

C.1.2 AI Chat dengan Respons dan Rekomendasi Konten



Gambar 3.6. Wireframe (Low-Fidelity) AI Chat dengan respons dan rekomendasi konten

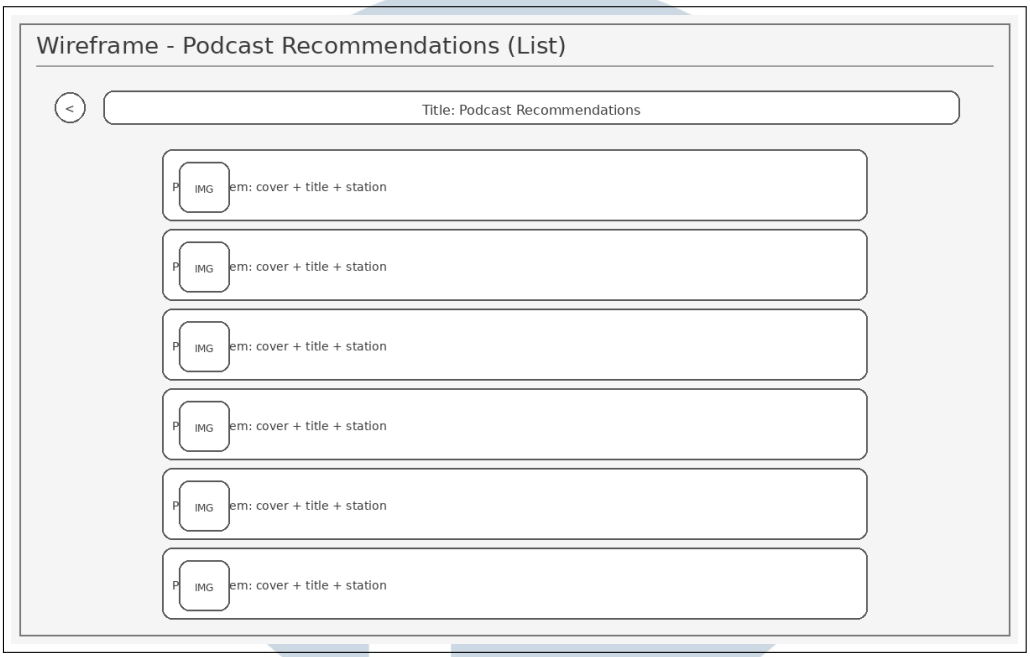
Gambar 3.6 menggambarkan kondisi ketika pengguna telah mengirim permintaan. Bubble chat pengguna ditempatkan di sisi kanan, sedangkan respons AI berada di sisi kiri. Di bawah respons AI, disediakan blok rekomendasi berupa list/kartu konten yang dapat diklik untuk membuka detail atau langsung diputar.



Gambar 3.7. Desain (High-Fidelity) Figma AI Chat dengan respons dan rekomendasi konten

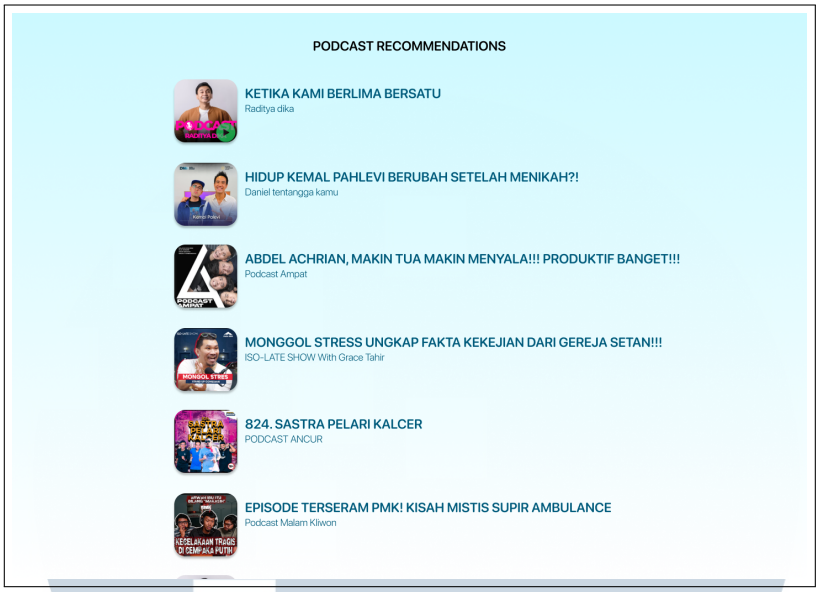
Gambar 3.7 menunjukkan tampilan high-fidelity AI Chat ketika AI memberikan respons berupa teks dan rekomendasi konten. Desain menonjolkan hierarki visual (bubble pengguna lebih kontras) dan menempatkan kartu rekomendasi tetap dekat dengan respons AI agar alur percakapan tidak terputus.

C.1.3 Halaman Podcast Recommendations (Web)



Gambar 3.8. Wireframe (Low-Fidelity) halaman Podcast Recommendations versi web

Gambar 3.8 menunjukkan wireframe halaman Podcast Recommendations yang menampilkan daftar podcast dalam format list. Setiap baris memuat cover, judul podcast, dan informasi singkat stasiun/kanal sehingga pengguna dapat memilih konten dengan cepat.



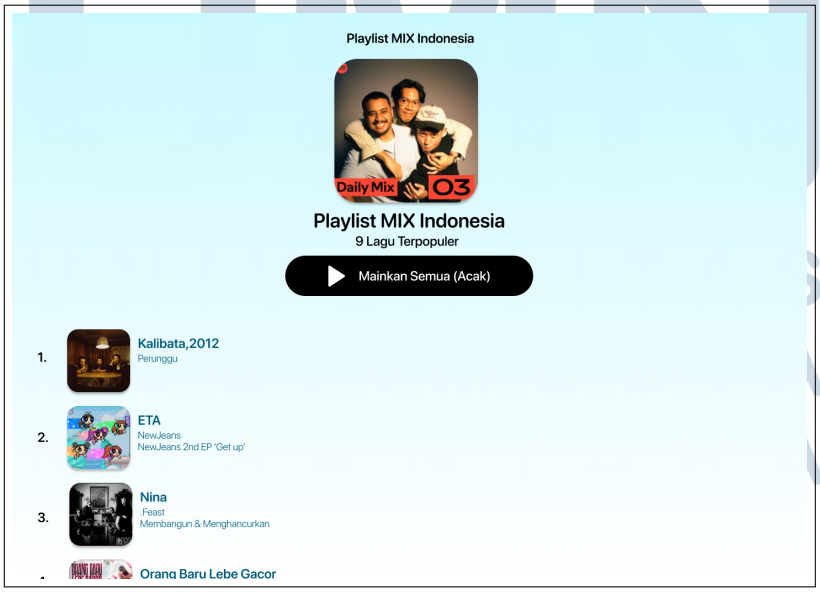
Gambar 3.9. Desain (High-Fidelity) Figma halaman Podcast Recommendations versi web

Gambar 3.9 memperlihatkan desain final halaman Podcast Recommendations pada Figma. Kontras teks, ukuran cover, dan spasi antar item dibuat agar daftar tetap nyaman dibaca meskipun jumlah konten banyak.

C.1.4 Halaman Top Playlist / Top Artist Detail (Web)

Gambar 3.10. Wireframe (Low-Fidelity) halaman Top Playlist / Top Artist Detail versi web

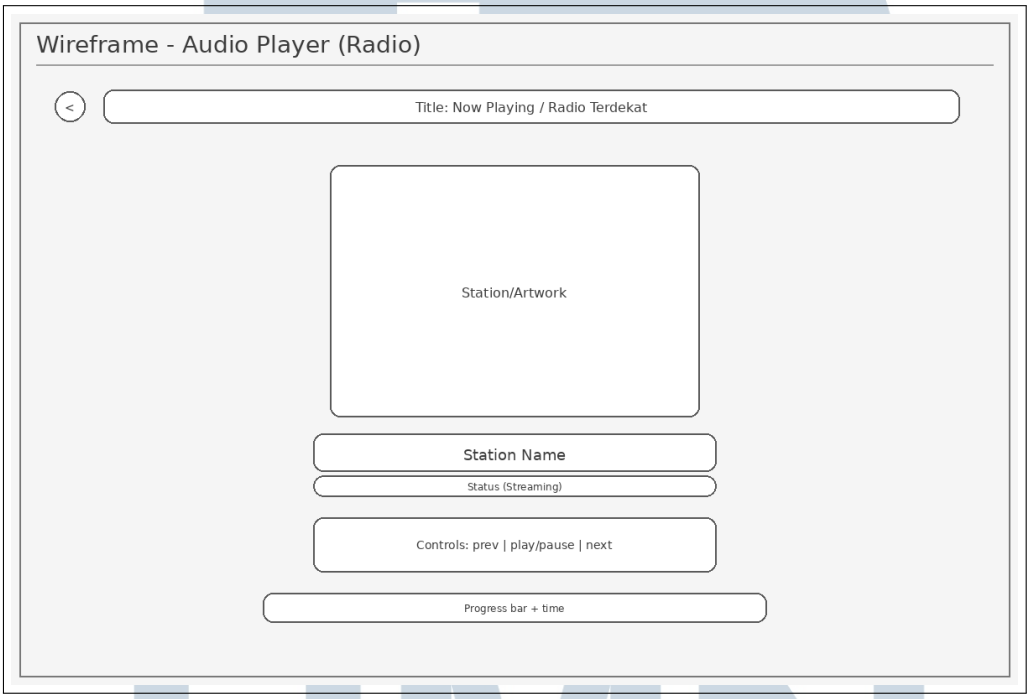
Gambar 3.10 memperlihatkan struktur halaman detail playlist/artist. Bagian atas menampilkan cover besar dan informasi meta (judul, genre), diikuti tombol aksi utama untuk memutar semua lagu, kemudian daftar trek pada bagian bawah.



Gambar 3.11. Desain (High-Fidelity) Figma halaman Top Playlist / Top Artist versi web

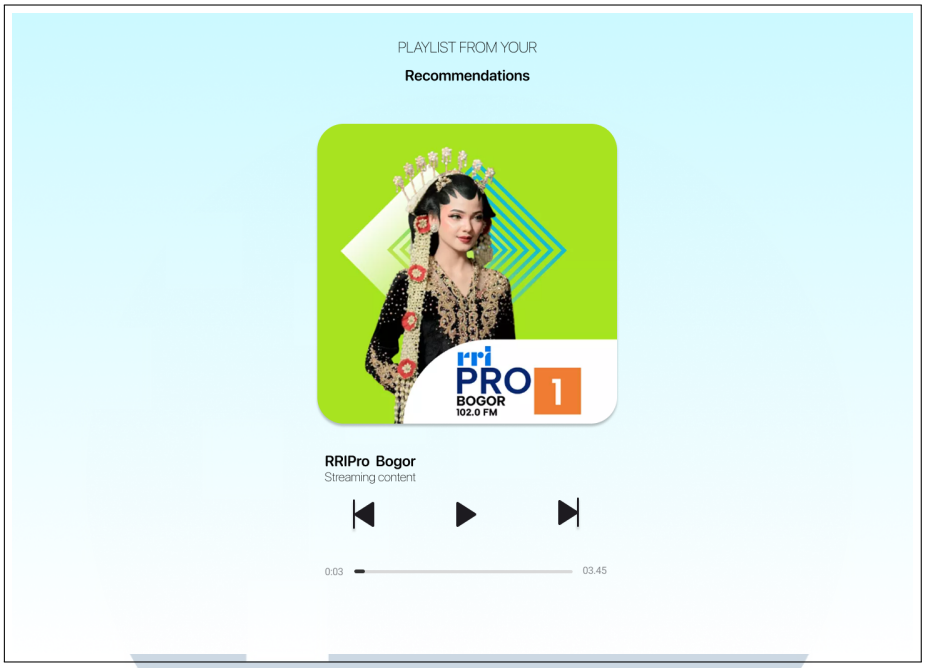
Gambar 3.11 menampilkan desain high-fidelity halaman playlist/artist. Tata letak menekankan cover sebagai fokus utama, tombol *Mainkan semua (acak)* sebagai aksi utama, serta daftar lagu yang tersusun rapi untuk memudahkan eksplorasi.

C.1.5 Halaman Audio Player Radio (Web)



Gambar 3.12. Wireframe (Low-Fidelity) halaman Audio Player Radio versi web

Gambar 3.12 menggambarkan layout pemutar radio dengan fokus pada identitas stasiun, status streaming, kontrol pemutaran, serta progress bar. Struktur dibuat minimalis untuk mengurangi distraksi sehingga pengguna lebih mudah mengontrol audio.



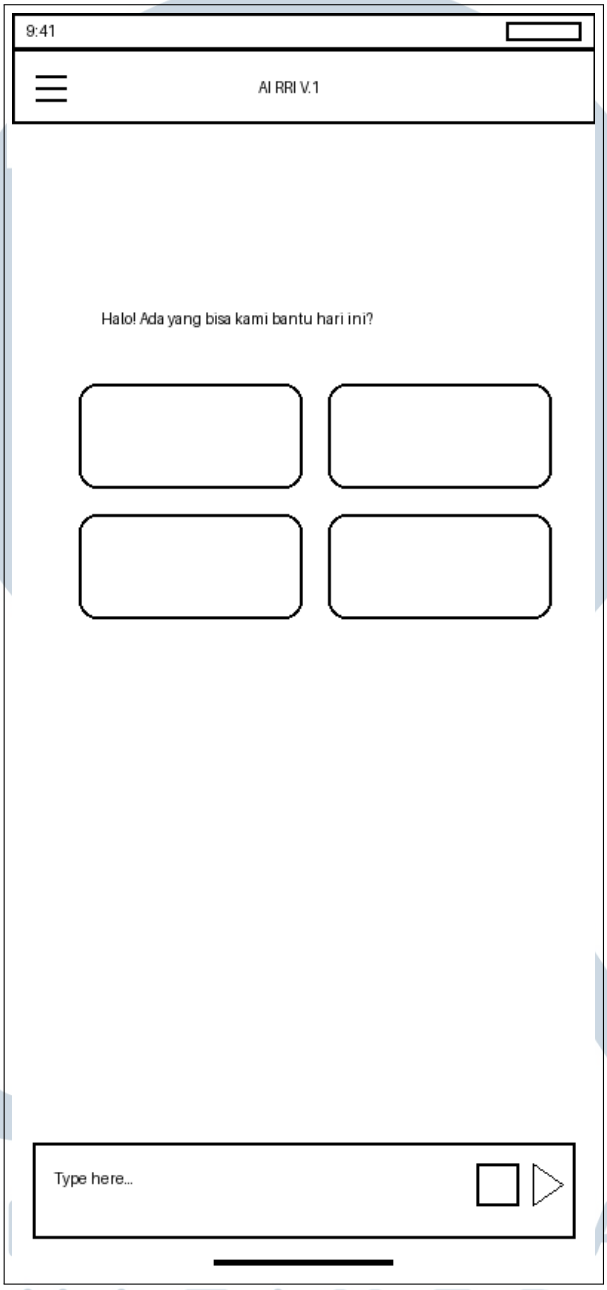
Gambar 3.13. Desain (High-Fidelity) Figma halaman Audio Player Radio versi web

Gambar 3.13 menunjukkan desain final pemutar radio pada Figma, termasuk konsistensi ukuran tombol kontrol, posisi progress bar, serta indikator status *streaming* agar informasi pemutaran mudah dipahami.

C.2 Perancangan Wireframe Versi Mobile

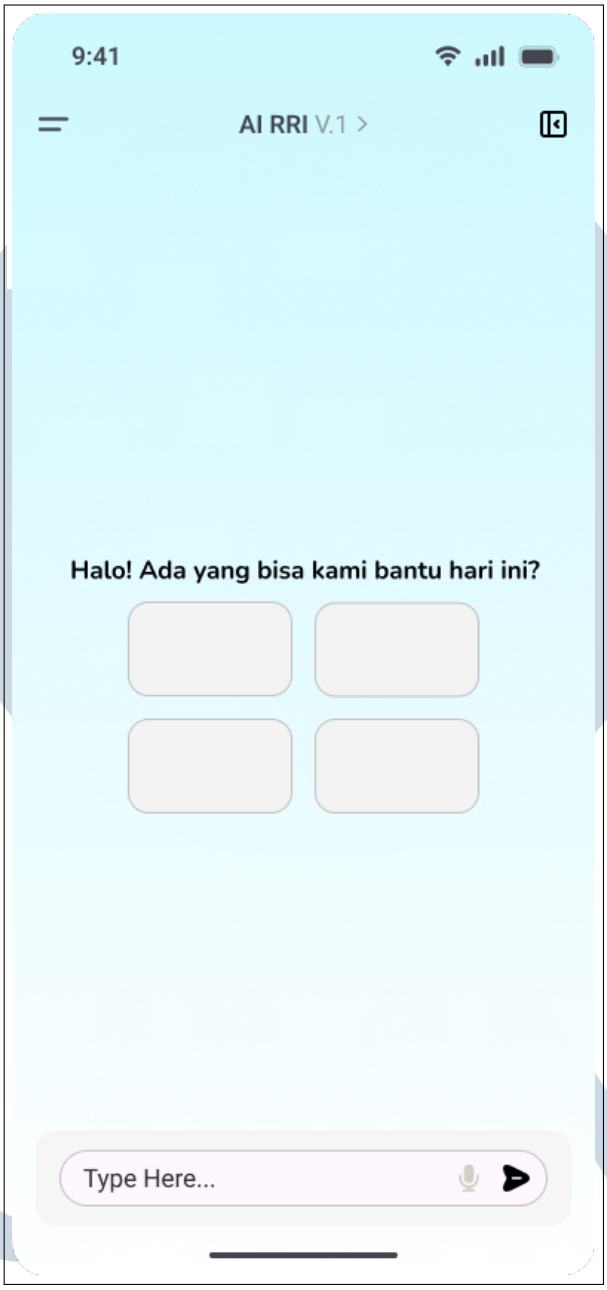
Untuk memastikan pengalaman pengguna tetap konsisten pada perangkat seluler, dibuat rancangan wireframe low-fidelity dan desain high-fidelity versi mobile. Struktur utama mengikuti versi web, namun disesuaikan dengan keterbatasan lebar layar, pola jangkauan jari, serta navigasi berbasis *drawer*.

C.2.1 AI Chat (Mobile) - Tampilan Awal New Chat



Gambar 3.14. Wireframe (Low-Fidelity) halaman AI Chat (awal *new chat*) versi mobile

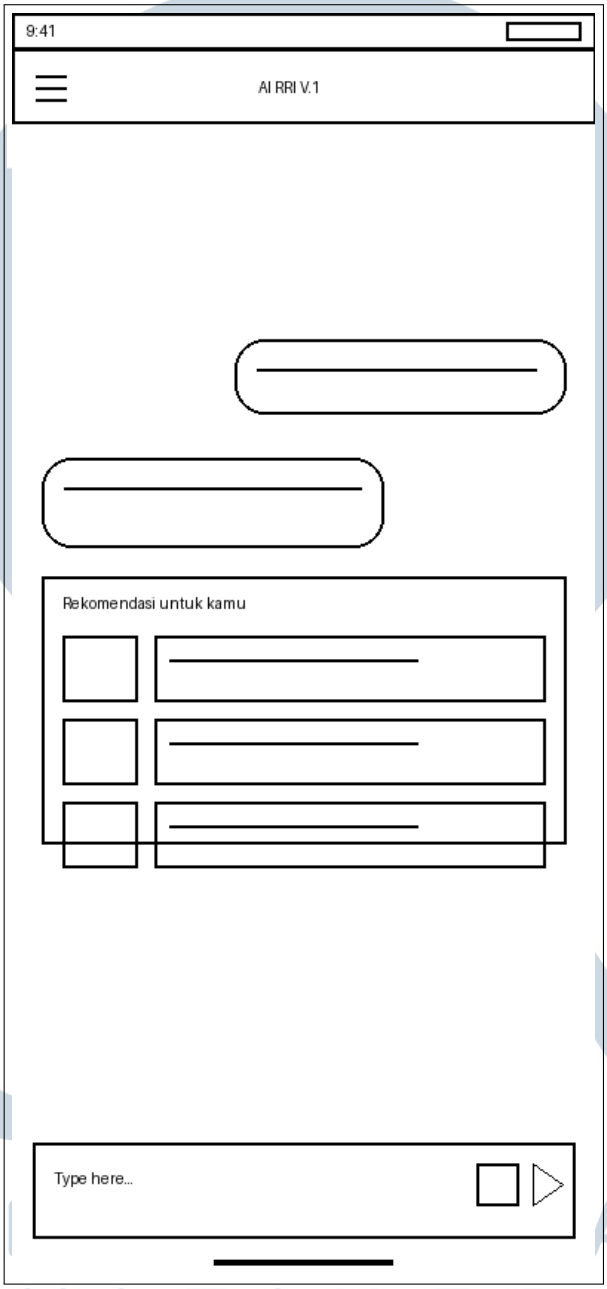
Gambar 3.14 memperlihatkan layout awal AI Chat pada perangkat mobile. Di bagian tengah terdapat *prompt card* sebagai pintasan pertanyaan, sedangkan di bagian bawah terdapat kolom input teks yang tetap (*sticky*) agar mudah dijangkau.



Gambar 3.15. Desain (High-Fidelity) Figma AI Chat (awal *new chat*) versi mobile

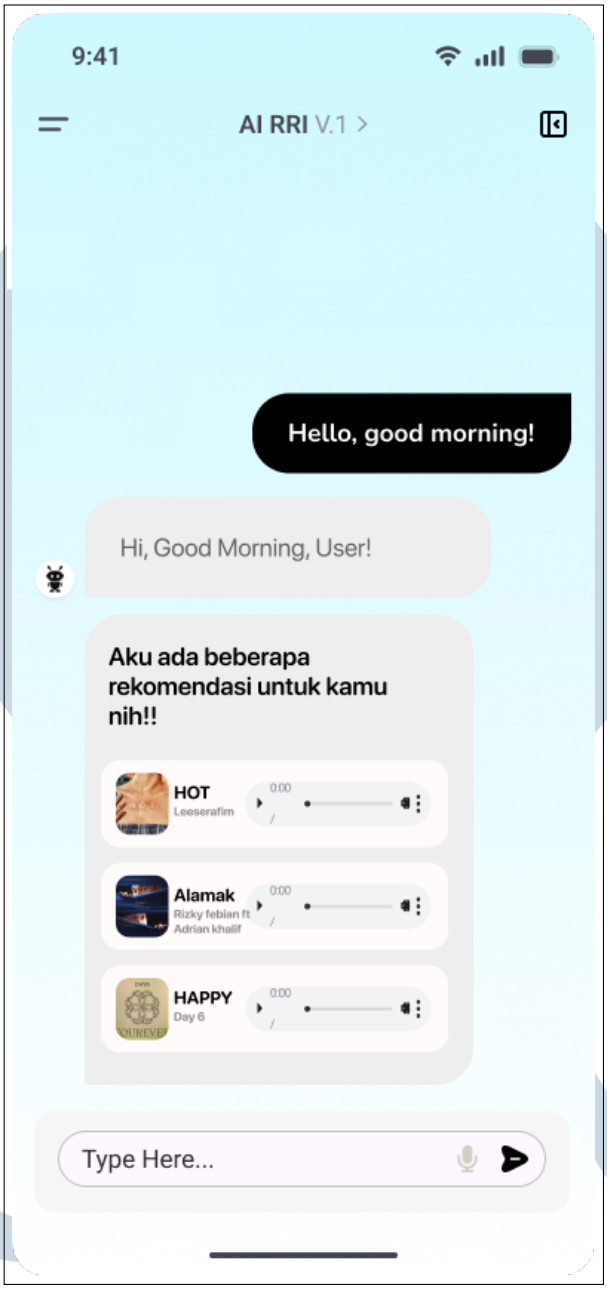
Gambar 3.15 merupakan desain high-fidelity dari tampilan awal AI Chat mobile. *Prompt card* dibuat lebih jelas sebagai elemen interaktif, dan input teks ditempatkan di bawah agar mendukung pola penggunaan satu tangan.

C.2.2 AI Chat (Mobile) - Percakapan dan Respons AI



Gambar 3.16. Wireframe (Low-Fidelity) tampilan percakapan AI Chat versi mobile

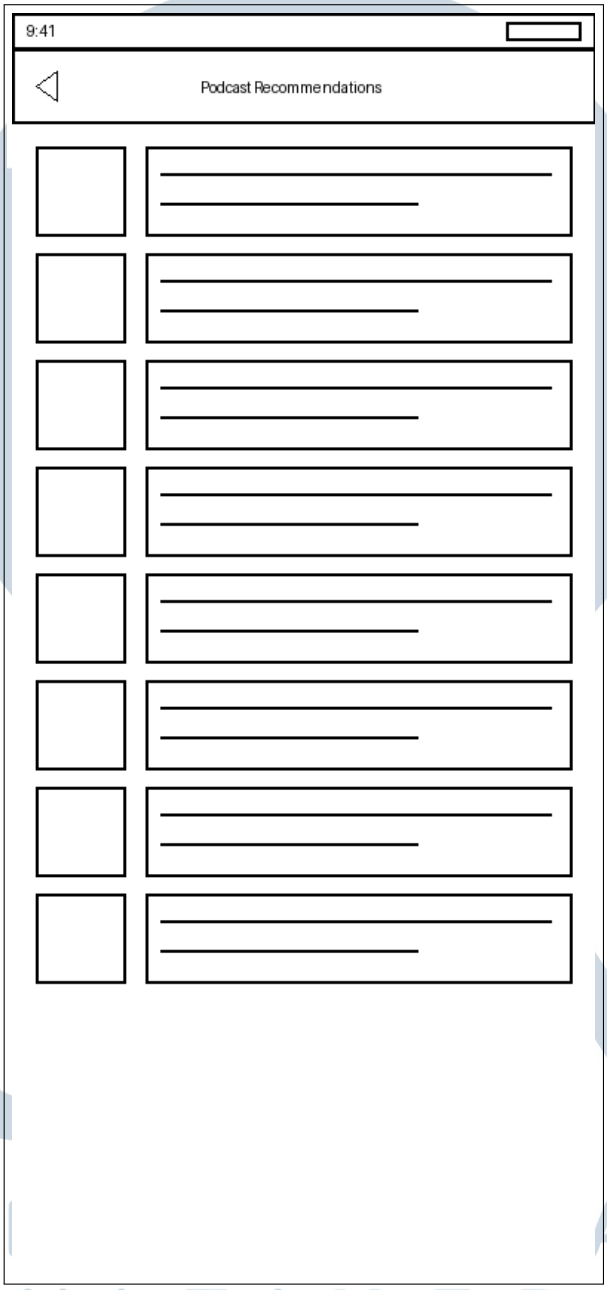
Gambar 3.16 menunjukkan struktur bubble percakapan di mobile, dengan pesan pengguna di sisi kanan dan respons AI di sisi kiri. Kolom input tetap berada di bagian bawah layar untuk mempertahankan konsistensi interaksi.



Gambar 3.17. Desain (High-Fidelity) Figma tampilan percakapan AI Chat versi mobile

Gambar 3.17 menunjukkan implementasi high-fidelity percakapan AI Chat pada mobile, dengan hierarki bubble yang jelas (bubble pengguna lebih kontras) agar pengguna mudah membedakan pesan.

C.2.3 Podcast Recommendations (Mobile)



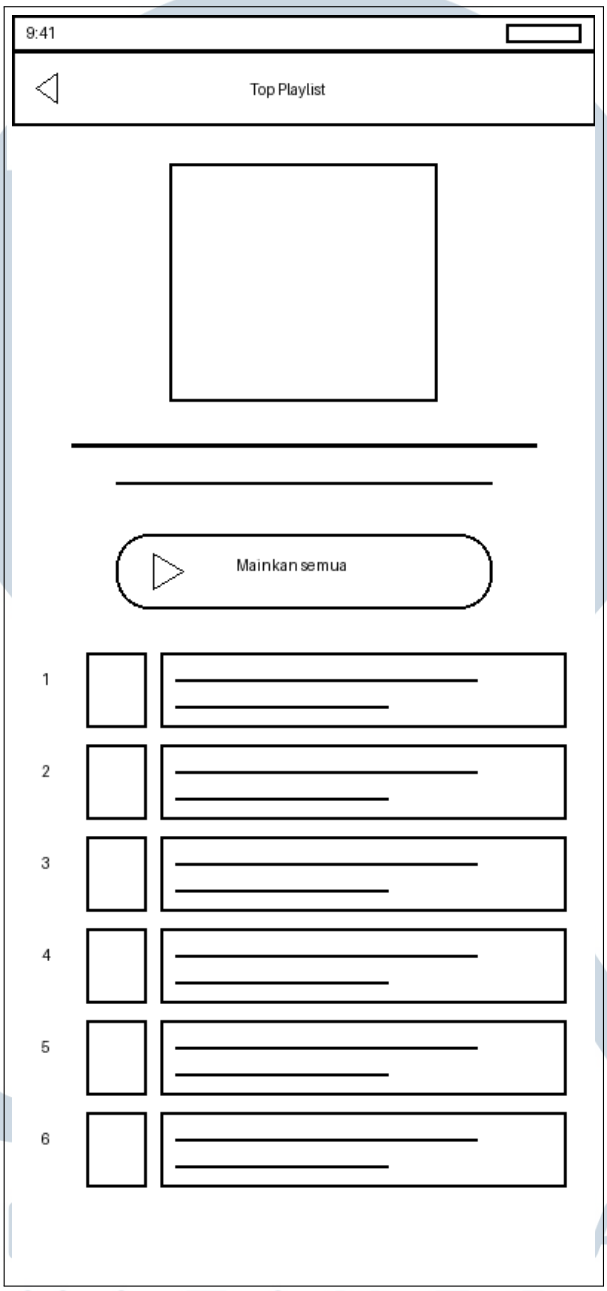
Gambar 3.18. Wireframe (Low-Fidelity) halaman Podcast Recommendations versi mobile

Gambar 3.18 menggambarkan layout daftar podcast secara vertikal yang dapat discroll. Setiap item memuat thumbnail, judul, dan nama stasiun untuk memudahkan pemilihan konten.

Gambar 3.19. Desain (High-Fidelity) Figma halaman Podcast Recommendations versi mobile

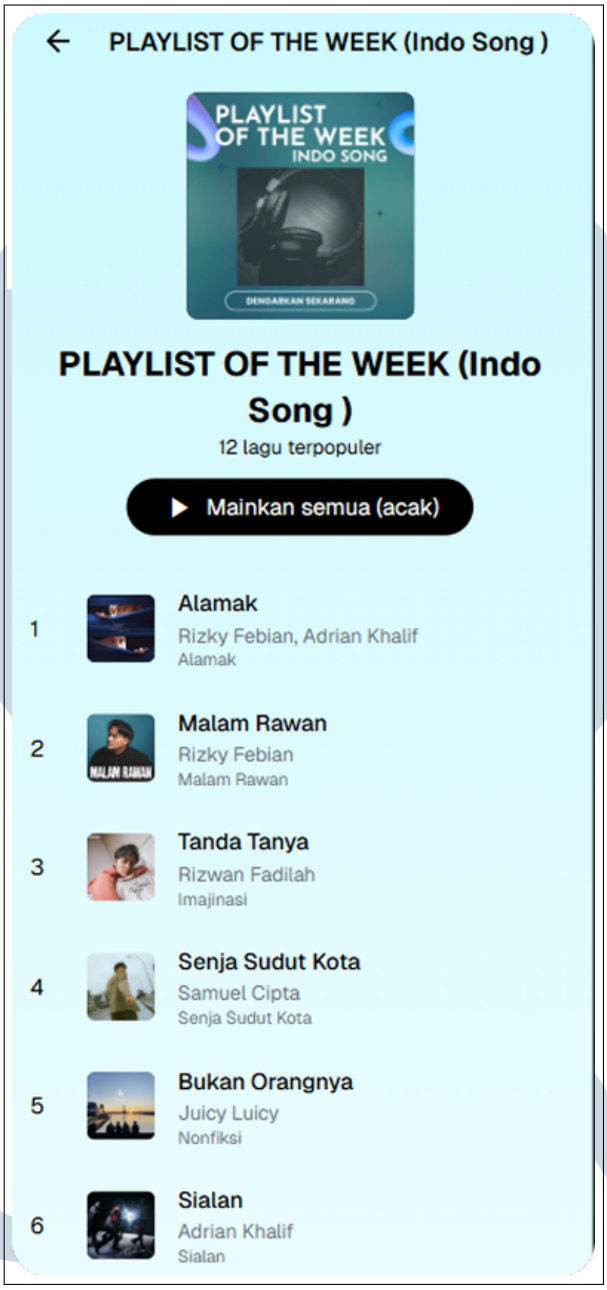
Gambar 3.19 menampilkan desain final daftar podcast pada mobile. Ukuran teks dibuat tegas dan item dibuat rapih agar tetap terbaca pada layar kecil.

C.2.4 Playlist Detail / Music Charts (Mobile)



Gambar 3.20. Wireframe (Low-Fidelity) halaman Playlist Detail versi mobile

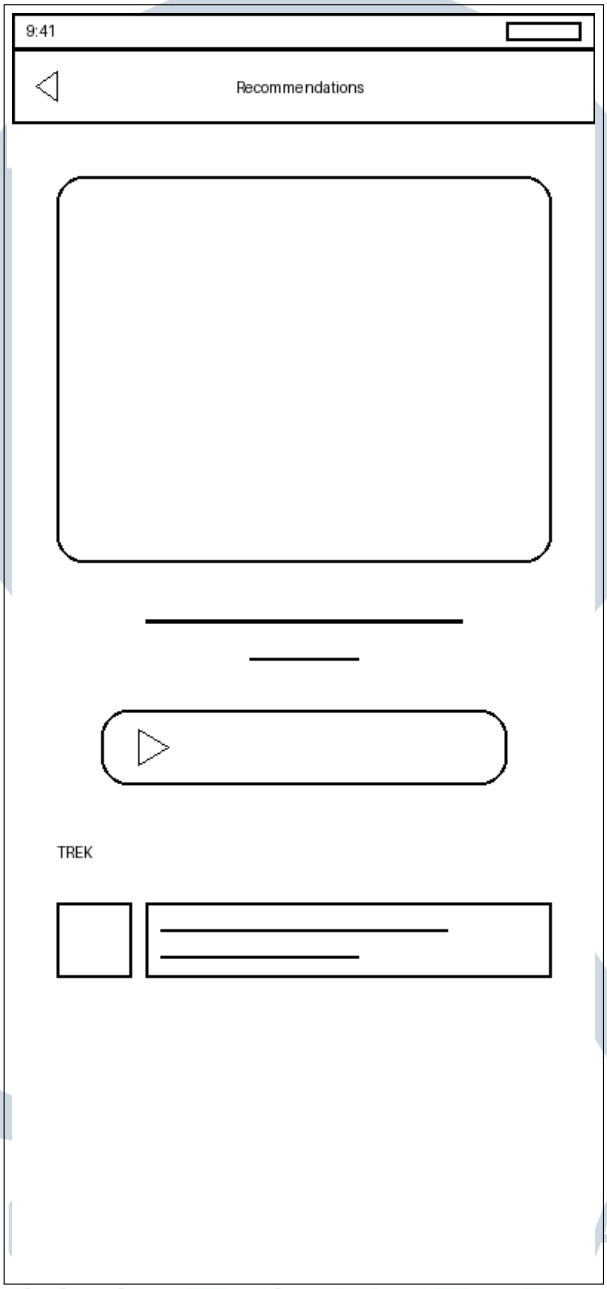
Gambar 3.20 memperlihatkan struktur halaman playlist dengan cover di bagian atas, informasi jumlah lagu, tombol *Mainkan semua*, dan daftar trek di bawahnya.



Gambar 3.21. Desain (High-Fidelity) Figma halaman Playlist Detail versi mobile

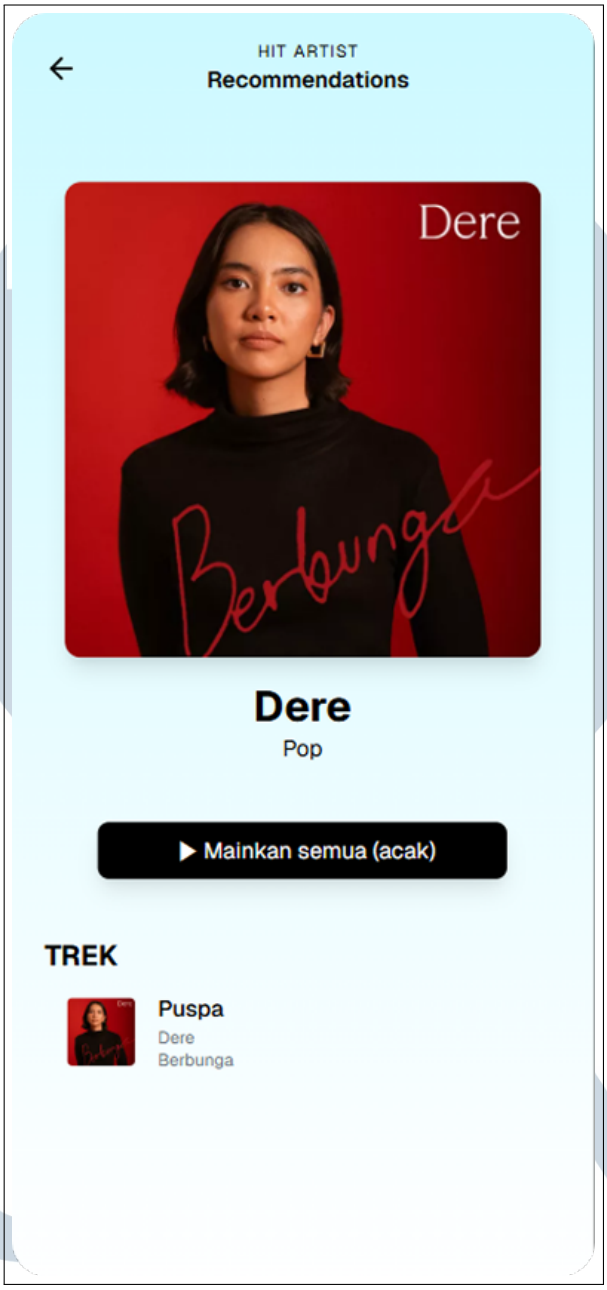
Gambar 3.21 menunjukkan desain high-fidelity playlist detail pada mobile, dengan tombol aksi utama dibuat lebar agar mudah ditekan serta daftar lagu ditampilkan dalam format vertikal yang nyaman discroll.

C.2.5 Hit Artist Detail (Mobile)



Gambar 3.22. Wireframe (Low-Fidelity) halaman Hit Artist versi mobile

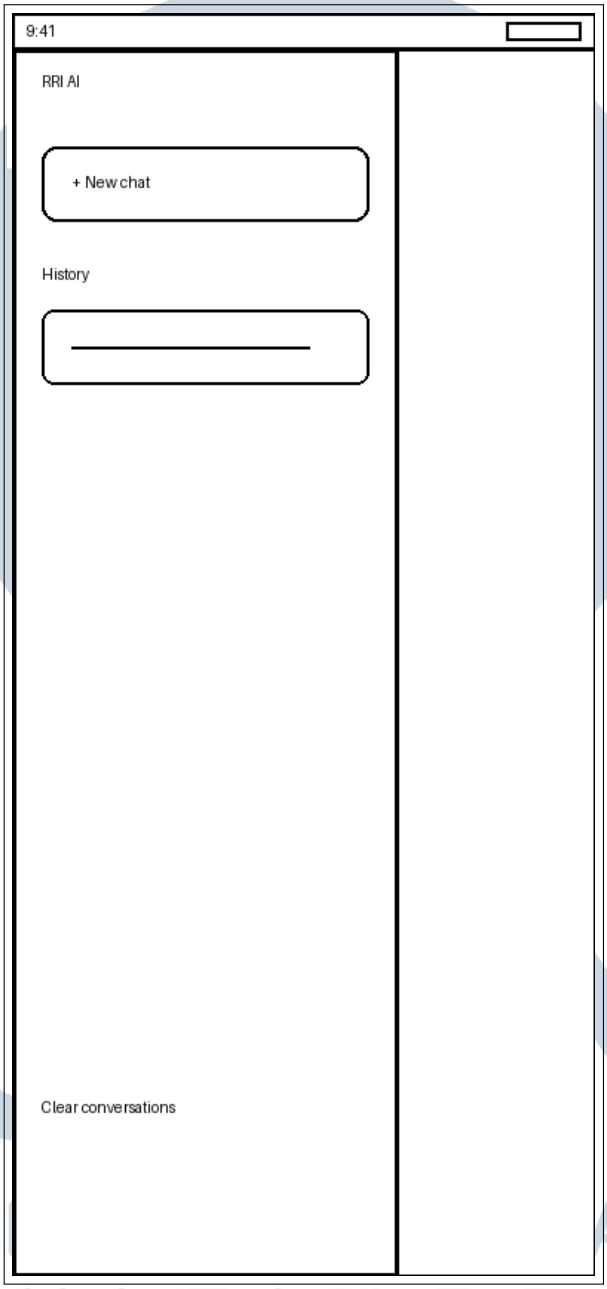
Gambar 3.22 menunjukkan layout halaman hit artist dengan fokus pada foto/cover artist, nama, genre, serta daftar trek yang direkomendasikan.



Gambar 3.23. Desain (High-Fidelity) Figma halaman Hit Artist versi mobile

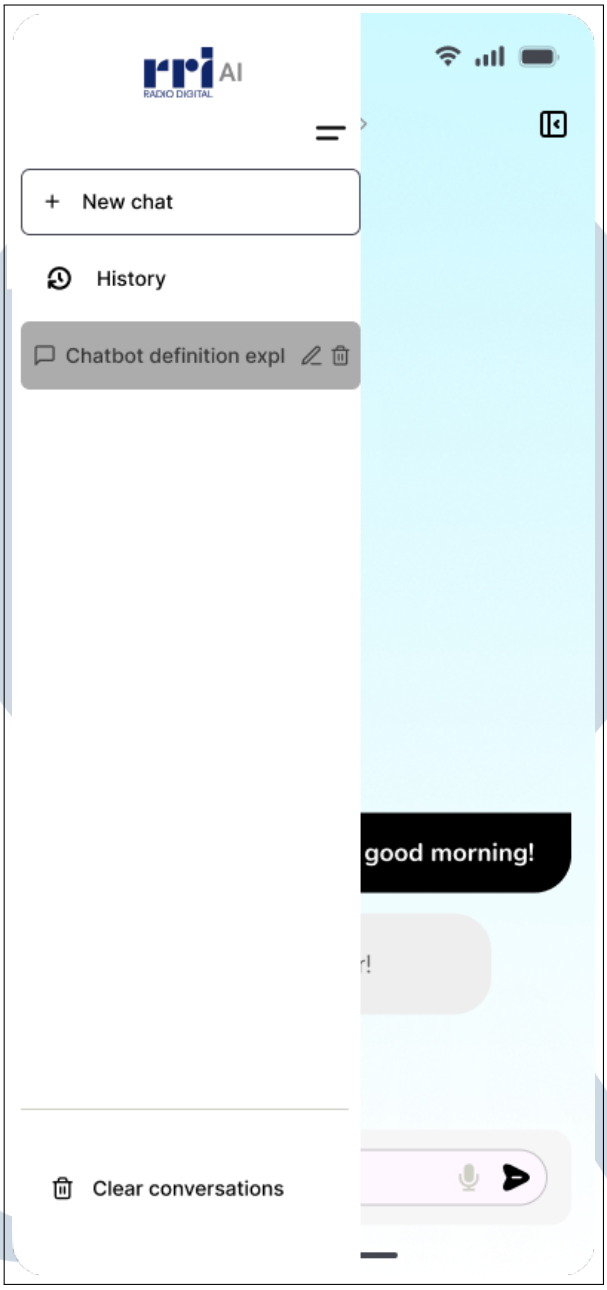
Gambar 3.23 merupakan desain high-fidelity untuk halaman Hit Artist. Fokus visual diberikan pada cover artist, sementara daftar trek ditampilkan ringkas agar pengguna dapat langsung memutar lagu.

C.2.6 Sidebar Drawer (Mobile)



Gambar 3.24. Wireframe (Low-Fidelity) tampilan sidebar navigasi (*drawer*) pada mobile

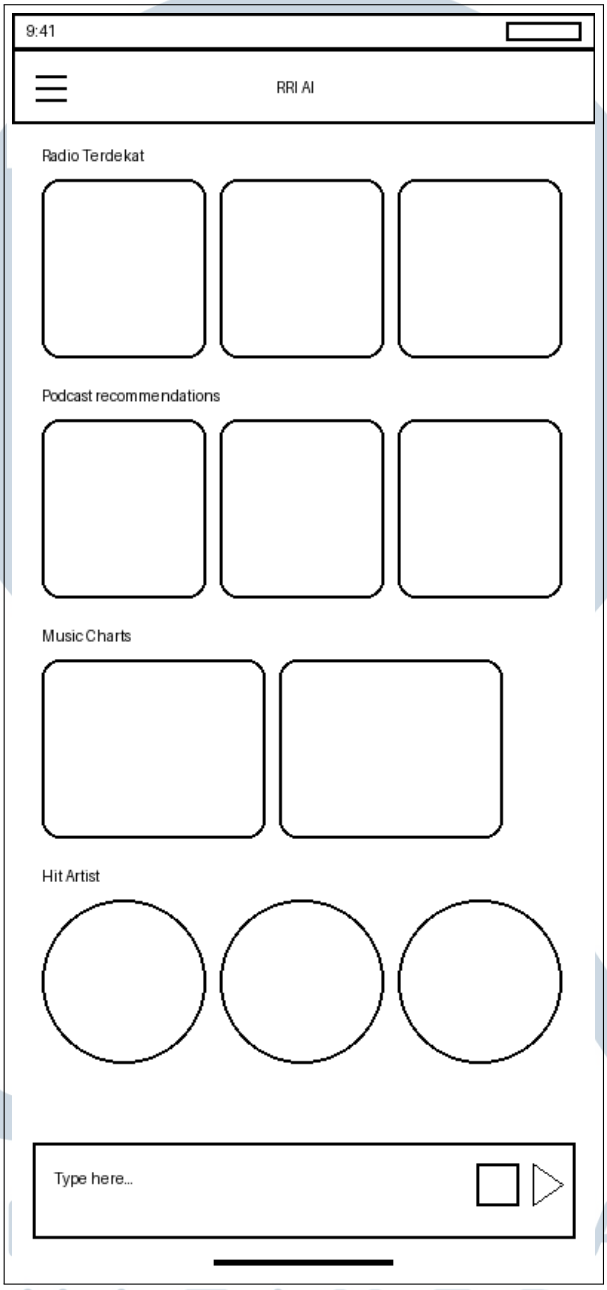
Gambar 3.24 menggambarkan pola navigasi dengan *drawer* dari sisi kiri, berisi tombol *New chat*, daftar riwayat percakapan, dan aksi *Clear conversations*.



Gambar 3.25. Desain (High-Fidelity) Figma sidebar navigasi (*drawer*) pada mobile

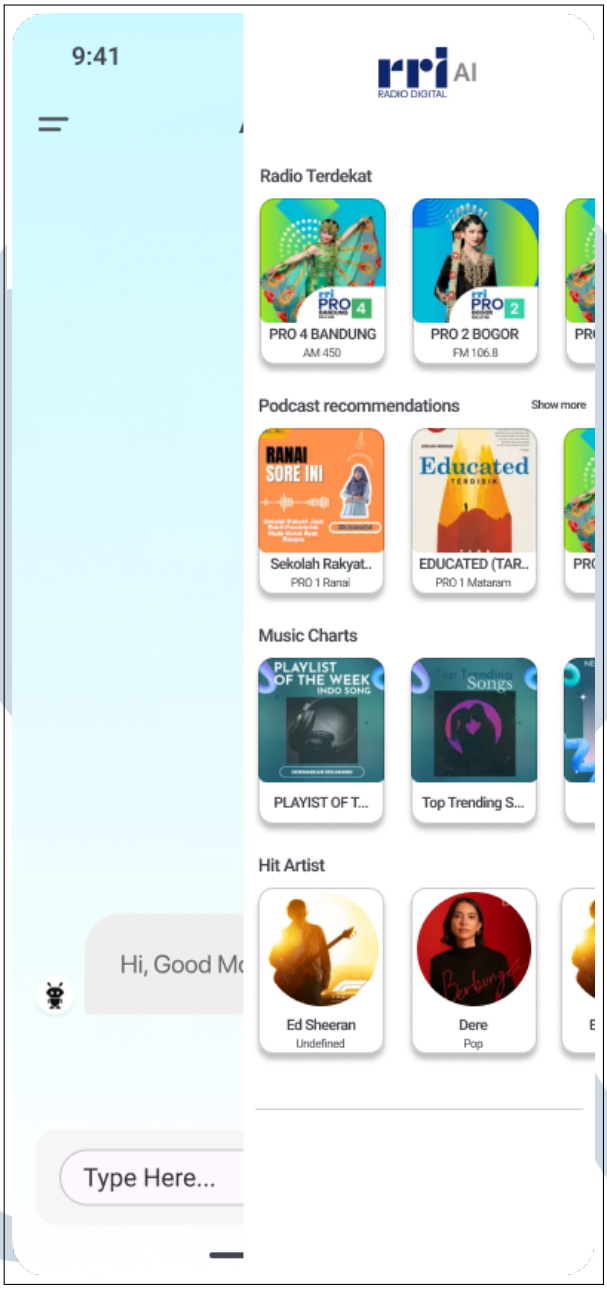
Gambar 3.25 menunjukkan desain final drawer sidebar. Penempatan tombol dibuat jelas agar pengguna cepat berpindah konteks chat tanpa mengganggu layar utama.

C.2.7 Panel Discovery (Mobile)



Gambar 3.26. Wireframe (Low-Fidelity) panel discovery versi mobile

Gambar 3.26 menunjukkan panel discovery yang muncul dari sisi kanan, menampilkan kumpulan kartu Radio Terdekat, Podcast Recommendations, Music Charts, dan Hit Artist agar pengguna dapat menjelajah konten dengan cepat.



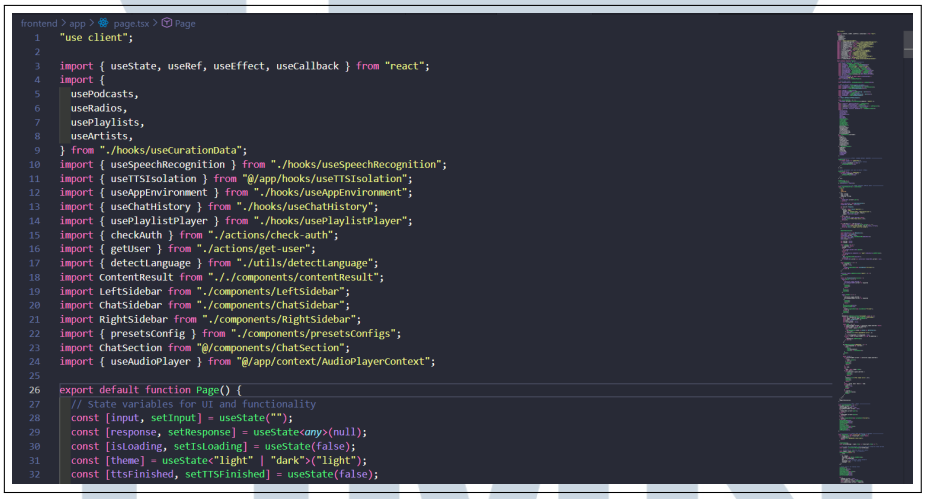
Gambar 3.27. Desain (High-Fidelity) Figma panel discovery versi mobile

Gambar 3.27 memperlihatkan desain high-fidelity panel discovery. Konten disajikan sebagai kartu-kartu yang dapat discroll/di-swipe sehingga eksplorasi konten lebih efisien pada layar kecil.

3.5 Implementasi Desain Menjadi Website Responsif

Setelah tahap perancangan wireframe dan desain Figma selesai, langkah berikutnya adalah mengimplementasikan desain tersebut menjadi aplikasi web responsif menggunakan *Next.js*, *TypeScript*, dan *Tailwind CSS*. Implementasi dilakukan dengan pendekatan komponen modular agar struktur kode mudah dirawat, serta memanfaatkan *global state/context* untuk fitur yang dipakai lintas halaman seperti pemutar audio. Pada bagian ini ditampilkan contoh potongan kode dan hasil tampilan pada perangkat web (desktop) serta contoh tampilan saat responsif pada perangkat mobile.

A. Implementasi Halaman Home / AI Chat

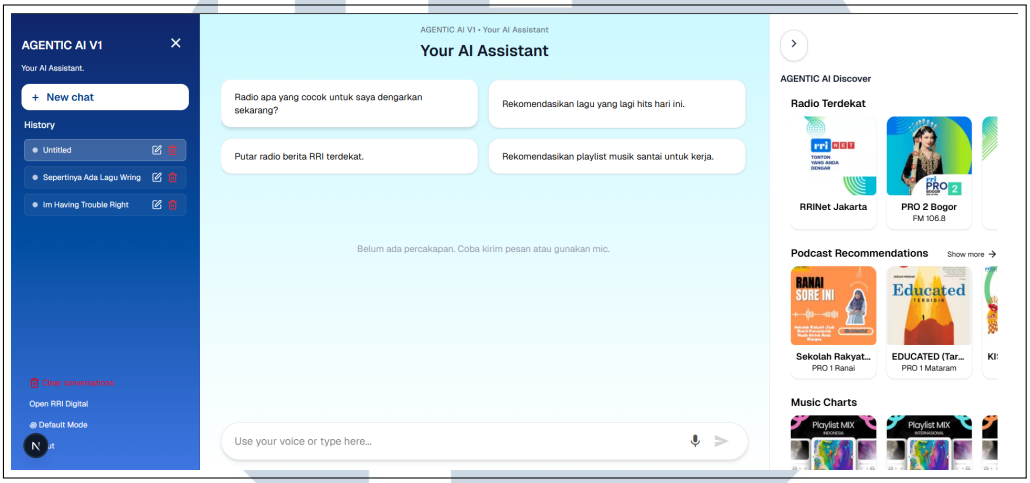


Gambar 3.28. Cuplikan kode komponen halaman Home / AI Chat

Gambar 3.28 memperlihatkan struktur utama halaman Home/AI Chat yang dibangun dengan komposisi beberapa komponen, seperti *LeftSidebar*, *ChatSection*, dan *RightSidebar*. Pada bagian ini juga terlihat penggunaan berbagai *custom hook* untuk mengambil data kurasi (*podcast, radio, playlist, artist*), mengelola riwayat chat, mendeteksi bahasa, hingga mengaktifkan *speech recognition* dan *text-to-speech*. Penggunaan hook tersebut bertujuan agar logika bisnis tidak menumpuk di komponen halaman, melainkan dipisahkan menjadi modul yang lebih kecil dan mudah diuji.

Pada implementasinya, halaman Home/AI Chat dibuat menyerupai layout desain Figma, yaitu pola tiga kolom pada desktop. Sidebar kiri digunakan untuk

navigasi dan riwayat percakapan, area tengah berperan sebagai area percakapan utama, dan panel kanan menampilkan *discovery* konten. Selain itu, terlihat juga penggunaan *AudioPlayerContext* untuk menghubungkan halaman Home dengan status pemutar audio global, sehingga ketika pengguna memutar konten dari rekomendasi, status pemutaran dapat konsisten meskipun berpindah halaman.



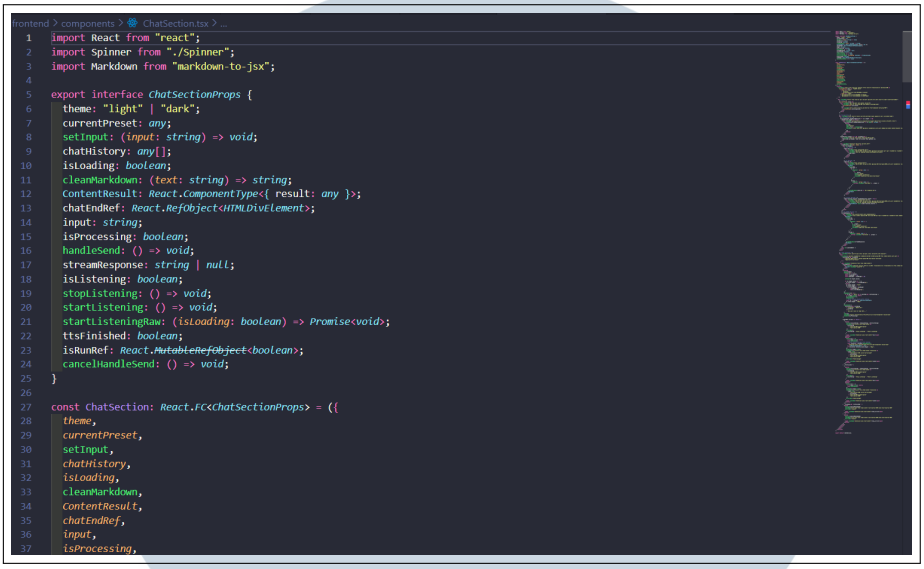
Gambar 3.29. Hasil implementasi halaman Home / AI Chat pada website (desktop)

Gambar 3.29 menunjukkan hasil implementasi halaman Home/AI Chat pada tampilan desktop. Struktur tiga kolom berhasil mengikuti rancangan desain, sehingga pengguna dapat berinteraksi dengan AI di area tengah dan tetap melihat rekomendasi konten pada panel discovery di sisi kanan.

Gambar 3.30. Hasil implementasi halaman Home / AI Chat pada tampilan mobile (responsif)

Gambar 3.30 menunjukkan tampilan halaman Home/AI Chat ketika diakses pada perangkat mobile. Pada mode responsif, elemen sidebar dan panel discovery disederhanakan agar tidak memakan lebar layar, sedangkan area chat tetap menjadi fokus utama.

B. Implementasi AI Chat dengan Respons Rekomendasi

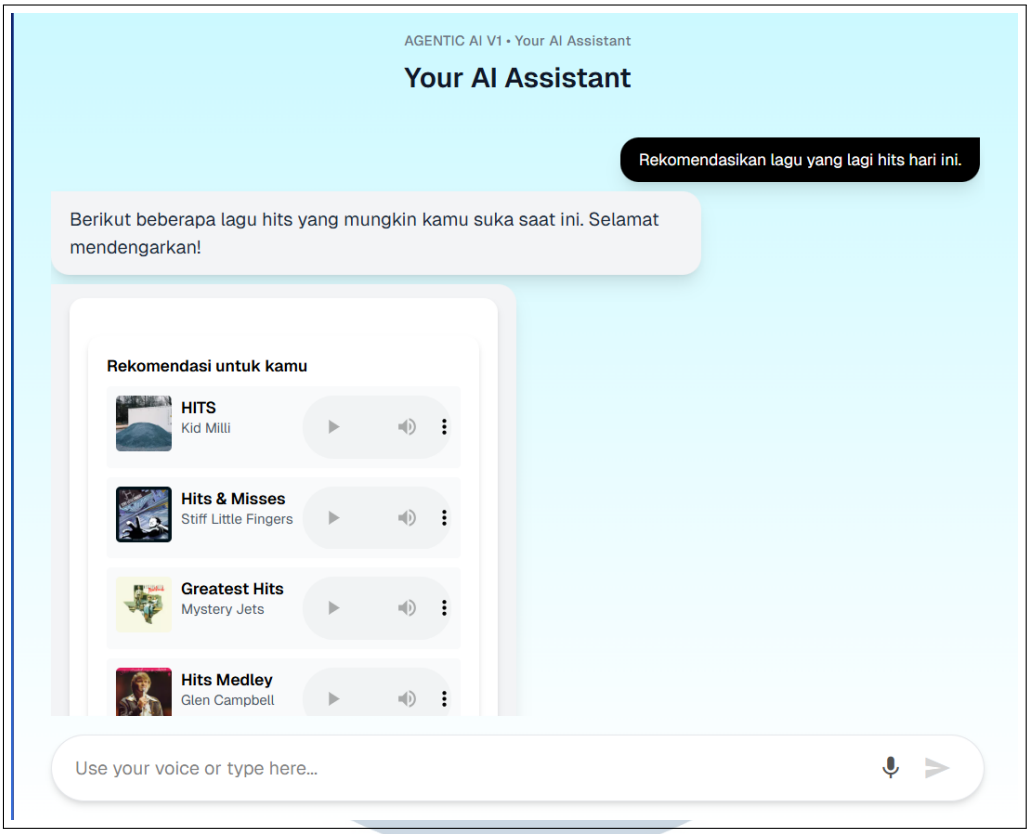


```
frontend > components > ChatSection.tsx > ...
1   import React from "react";
2   import Spinner from "./Spinner";
3   import Markdown from "markdown-to-jsx";
4
5   export interface ChatSectionProps {
6     theme: "light" | "dark";
7     currentPreset: any;
8     setInput: (input: string) => void;
9     chatHistory: any[];
10    isLoading: boolean;
11    cleanMarkdown: (text: string) => string;
12    ContentResult: React.ComponentType<{ result: any }>;
13    chatEndRef: React.RefObject<HTMLDivElement>;
14    input: string;
15    isProcessing: boolean;
16    handleSend: () => void;
17    streamResponse: string | null;
18    isListening: boolean;
19    stopListening: () => void;
20    startListening: () => void;
21    startListeningRaw: (isLoading: boolean) => Promise<void>;
22    ttsFinished: boolean;
23    isRunRef: React.MutableRefObject<boolean>;
24    cancelHandleSend: () => void;
25  }
26
27  const ChatSection: React.FC<ChatSectionProps> = ({
28    theme,
29    currentPreset,
30    setInput,
31    chatHistory,
32    isLoading,
33    cleanMarkdown,
34    ContentResult,
35    chatEndRef,
36    input,
37    isProcessing,
```

Gambar 3.31. Cuplikan kode komponen AI Chat dan kartu rekomendasi

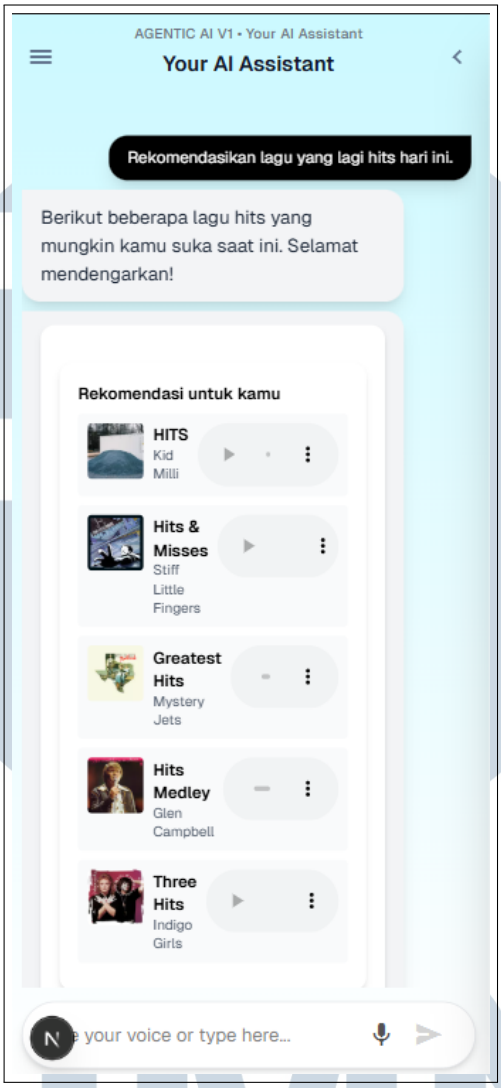
Gambar 3.31 menampilkan bagian implementasi AI Chat yang menangani alur input pengguna hingga render respons dari AI. Secara umum, potongan kode ini mengatur state untuk *input* dan *response*, mengelola status *loading* ketika menunggu respons dari backend, serta memastikan tampilan bubble chat konsisten berdasarkan peran (pesan pengguna dan pesan AI). Pada implementasi respons, data yang diterima dari backend dirender secara dinamis sehingga AI tidak hanya mengembalikan teks, tetapi juga dapat menampilkan konten rekomendasi seperti lagu/playlist/podcast yang relevan.

Pada bagian ini juga terlihat pemecahan struktur menjadi komponen-komponen UI, misalnya komponen untuk bubble chat, komponen hasil konten (*ContentResult*), dan komponen sidebar. Pendekatan modular ini memudahkan pengembangan lanjutan, seperti menambahkan tipe rekomendasi baru tanpa mengubah struktur utama chat.



Gambar 3.32. Hasil implementasi tampilan AI Chat dengan rekomendasi konten (desktop)

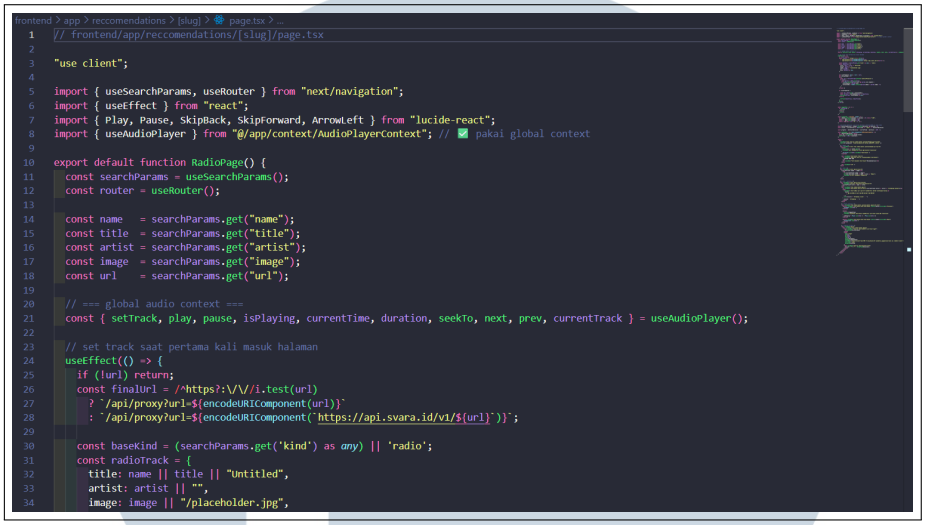
Gambar 3.32 menunjukkan AI Chat ketika AI memberikan rekomendasi konten. Bubble chat pengguna tampil lebih kontras, sedangkan respons AI muncul di sisi berlawanan agar hierarki pesan jelas. Rekomendasi konten ditampilkan dekat dengan respons AI sehingga pengguna dapat langsung melakukan aksi seperti membuka detail atau memutar konten.



Gambar 3.33. Hasil implementasi AI Chat pada tampilan mobile (responsif)

Gambar 3.33 menunjukkan tampilan AI Chat pada perangkat mobile. Tata letak disesuaikan menjadi satu kolom agar bubble chat tetap nyaman dibaca dan input chat mudah dijangkau pada bagian bawah layar.

C. Implementasi Halaman Podcast Recommendations

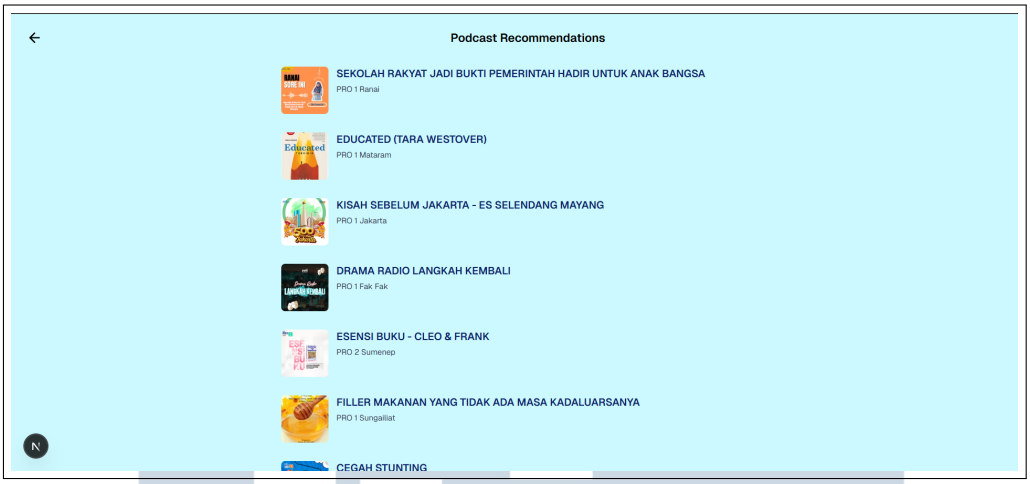


```
// frontend/app/reccomendations/[slug]/page.tsx

"use client";

import { useSearchParams, useRouter } from "next/navigation";
import { useEffect } from "react";
import { Play, Pause, SkipBack, SkipForward, ArrowLeft } from "lucide-react";
import { useAudioPlayer } from "@/app/context/AudioPlayerContext"; // ✅ pakai global context

export default function RadioPage() {
  const searchParams = useSearchParams();
  const router = useRouter();

  const name   = searchParams.get("name");
  const title  = searchParams.get("title");
  const artist = searchParams.get("artist");
  const image  = searchParams.get("image");
  const url    = searchParams.get("url");

  // === global audio context ===
  const { setTrack, play, pause, isPlaying, currentTime, duration, seekTo, next, prev, currentTrack } = useAudioPlayer();

  // set track saat pertama kali masuk halaman
  useEffect(() => {
    if (!url) return;
    const finalUrl = /^https?:\/\//i.test(url)
      ? `/api/proxy?url=${encodeURIComponent(url)}`
      : `/api/proxy?url=${encodeURIComponent(`https://api.svara.id/v1/${url}`)}`;

    const baseKind = (searchParams.get('kind') as any) || 'radio';
    const radioTrack = {
      title: name || title || "Untitled",
      artist: artist || "",
      image: image || "/placeholder.jpg",
```

Gambar 3.34. Cuplikan kode halaman Podcast Recommendations

Gambar 3.34 menunjukkan implementasi halaman Podcast Recommendations yang memanfaatkan data dari API untuk ditampilkan menjadi daftar podcast. Pada bagian ini umumnya terdapat proses pengambilan data, normalisasi struktur data agar konsisten untuk UI, serta pemetaan (*mapping*) item podcast ke komponen daftar/kartu. Struktur komponen dibuat berulang dan konsisten, sehingga setiap item memiliki elemen visual utama berupa cover, judul, dan informasi stasiun/kanal.

Dengan pola daftar seperti ini, performa dan keterbacaan tetap terjaga walaupun jumlah rekomendasi banyak, karena komponen yang dirender bersifat seragam dan mudah dikontrol melalui styling Tailwind CSS.



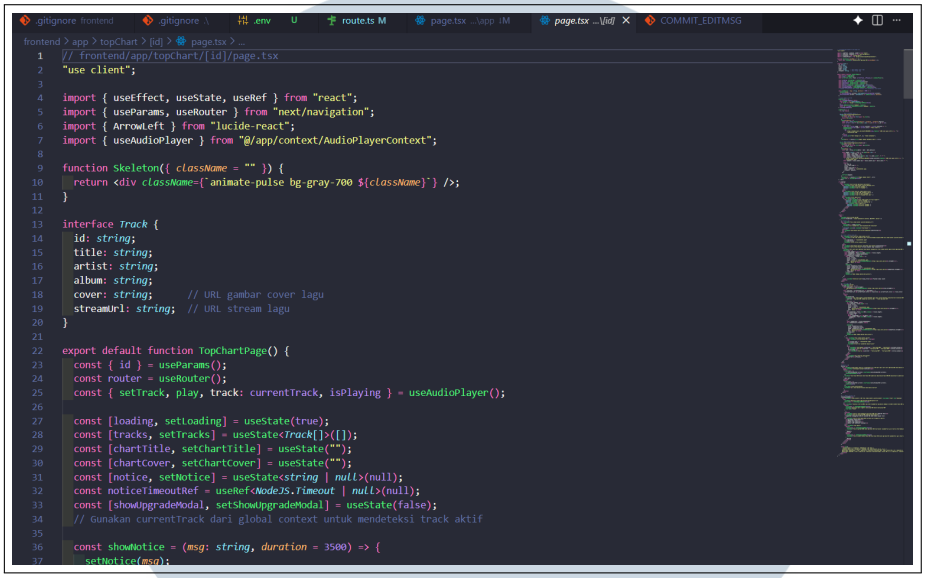
Gambar 3.35. Hasil implementasi halaman Podcast Recommendations pada website (desktop)

Gambar 3.35 memperlihatkan tampilan akhir halaman Podcast Recommendations pada web. Daftar podcast ditampilkan secara vertikal dengan struktur yang rapi sehingga pengguna dapat melakukan *scroll* untuk menelusuri rekomendasi.

Gambar 3.36. Hasil implementasi halaman Podcast Recommendations pada tampilan mobile (responsif)

Gambar 3.36 menunjukkan tampilan daftar podcast pada perangkat mobile. Elemen cover dan teks disusun agar tetap terbaca di layar kecil, sementara pola *scroll* vertikal dipertahankan untuk kemudahan eksplorasi.

D. Implementasi Halaman Top Playlist / Top Artist

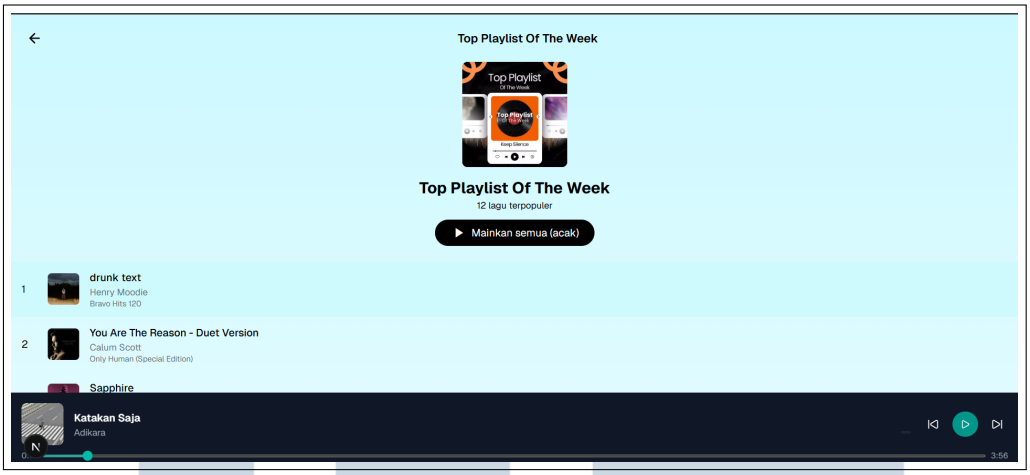


```tsx
// frontend/app/topChart/[id]/page.tsx
"use client";

import { useEffect, useState, useRef } from "react";
import { useParams, useRouter } from "next/navigation";
import { ArrowLeft } from "lucide-react";
import { useAudioPlayer } from "@/app/context/AudioPlayerContext";

function Skeleton({ className = "" }) {
  return <div className={`animate-pulse bg-gray-700 ${className}`} />;
}

interface Track {
  id: string;
  title: string;
  artist: string;
  album: string;
  cover: string;       // URL gambar cover lagu
  streamUrl: string;  // URL stream lagu
}

export default function TopChartPage() {
  const { id } = useParams();
  const router = useRouter();
  const { setTrack, play, track: currentTrack, isPlaying } = useAudioPlayer();

  const [loading, setLoading] = useState(true);
  const [tracks, setTracks] = useState<Track[]>([]);
  const [chartTitle, setChartTitle] = useState("");
  const [chartCover, setChartCover] = useState("");
  const [notice, setNotice] = useState<string | null>(null);
  const noticeTimeoutRef = useRef<NodeJS.Timeout | null>(null);
  const [showUpgradeModal, setShowUpgradeModal] = useState(false);
  // Gunakan currentTrack dari global context untuk mendeteksi track aktif

  const showNotice = (msg: string, duration = 3500) => {
    setNotice(msg);
```

Gambar 3.37. Cuplikan kode halaman Top Playlist / Top Artist

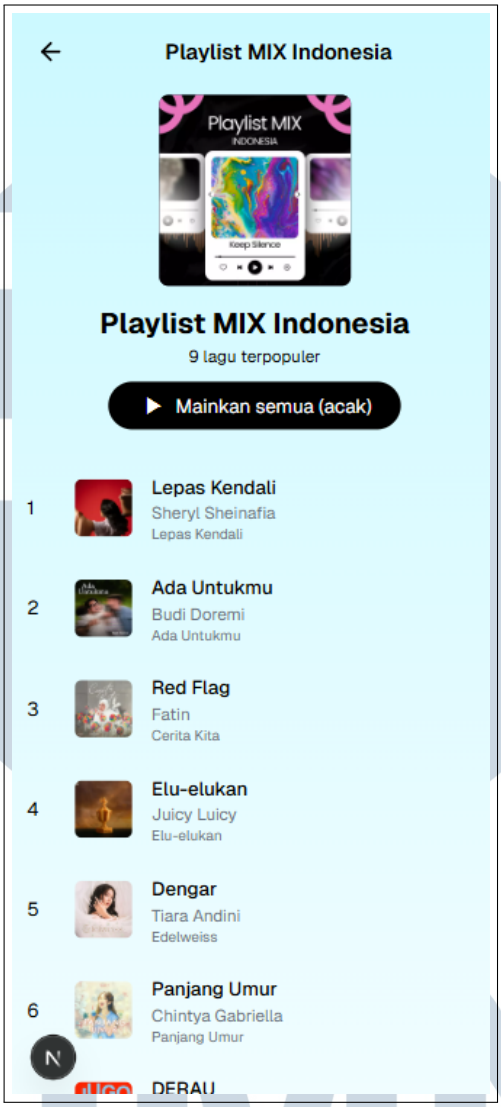
Gambar 3.37 memperlihatkan implementasi halaman detail playlist/artist, yang umumnya terdiri dari area cover utama, informasi metadata (judul, kategori/genre, jumlah trek), tombol aksi utama seperti *Mainkan semua (acak)*, serta daftar trek. Pada halaman ini, daftar trek biasanya dirender secara dinamis dari data API dan setiap item trek memiliki aksi untuk memutar lagu tertentu.

Di dalam kode juga terlihat keterhubungan dengan *global audio context*, sehingga ketika pengguna menekan tombol *play all* atau memilih satu trek, track aktif dapat ditetapkan dan dimainkan tanpa harus membuat logika pemutar audio terpisah di tiap halaman.



Gambar 3.38. Hasil implementasi halaman Top Playlist / Top Artist pada website (desktop)

Gambar 3.38 menunjukkan tampilan halaman playlist/artist yang telah diimplementasikan. Pengguna dapat memutar semua lagu atau memilih lagu tertentu melalui daftar trek yang disusun secara vertikal.

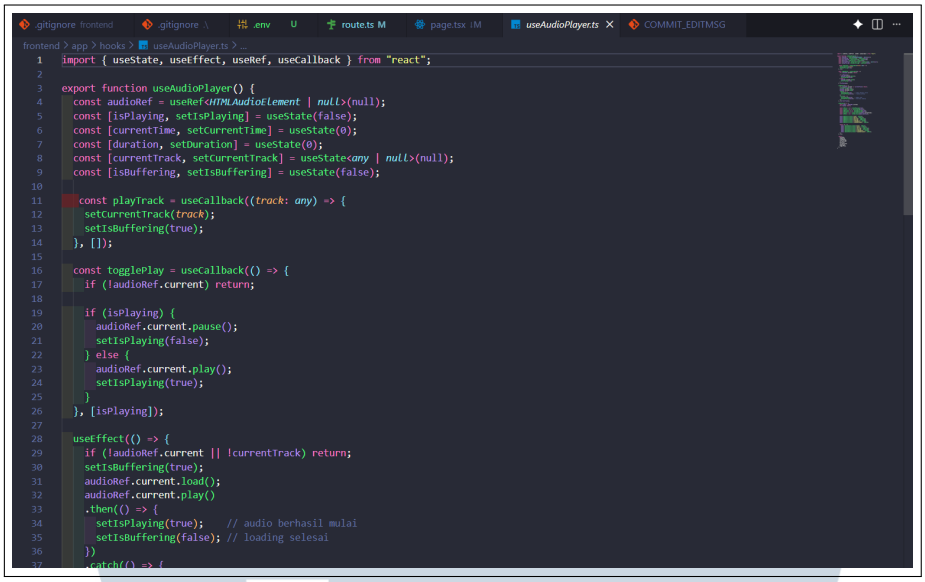


Gambar 3.39. Hasil implementasi halaman Playlist/Artist pada tampilan mobile (responsif)

Gambar 3.39 menunjukkan tampilan halaman playlist/artist pada perangkat mobile. Tombol aksi dibuat lebih lebar agar mudah ditekan, sedangkan daftar trek dioptimalkan untuk *scroll*.

E. Implementasi Halaman Audio Player Radio

Halaman Audio Player Radio diimplementasikan menggunakan Next.js dan TypeScript dengan memanfaatkan *global audio context* agar status pemutaran dapat dikelola secara konsisten di berbagai halaman. Implementasi ini menangani pemilihan sumber streaming, kontrol pemutaran, perubahan track, serta sinkronisasi progress audio.



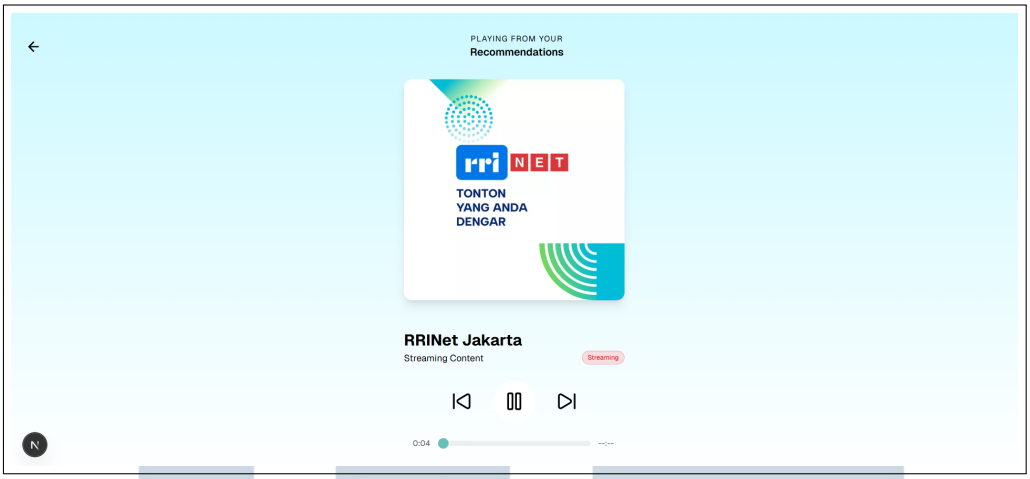
```
import { useState, useEffect, useRef, useCallback } from "react";

export function useAudioPlayer() {
  const audioRef = useRef<HTMLAudioElement | null>(null);
  const [isPlaying, setIsPlaying] = useState(false);
  const [currentTime, setCurrentTime] = useState(0);
  const [duration, setDuration] = useState(0);
  const [currentTrack, setCurrentTrack] = useState<any | null>(null);
  const [isBuffering, setIsBuffering] = useState(false);

  const playTrack = useCallback((track: any) => {
    setCurrentTrack(track);
    setIsBuffering(true);
  }, []);

  const togglePlay = useCallback(() => {
    if (!audioRef.current) return;

    if (isPlaying) {
      audioRef.current.pause();
      setIsPlaying(false);
    } else {
      audioRef.current.play();
      setIsPlaying(true);
    }
  }, [isPlaying]);

  useEffect(() => {
    if (!audioRef.current || !currentTrack) return;
    setIsBuffering(true);
    audioRef.current.load();
    audioRef.current.play()
    .then(() => {
      setIsPlaying(true);    // audio berhasil mulai
      setIsBuffering(false); // loading selesai
    })
```

Gambar 3.40. Cuplikan kode komponen Audio Player Radio

Gambar 3.40 menunjukkan implementasi halaman Radio Player (file `page.tsx`) yang mengambil parameter dari URL menggunakan `useSearchParams()` untuk membentuk metadata konten yang diputar, seperti `name`, `title`, `artist`, `image`, dan `url`. Setelah metadata terbentuk, halaman menghubungkannya ke *global audio context* melalui `useAudioPlayer()` agar kontrol pemutaran dapat konsisten dan dapat digunakan ulang di halaman lain.

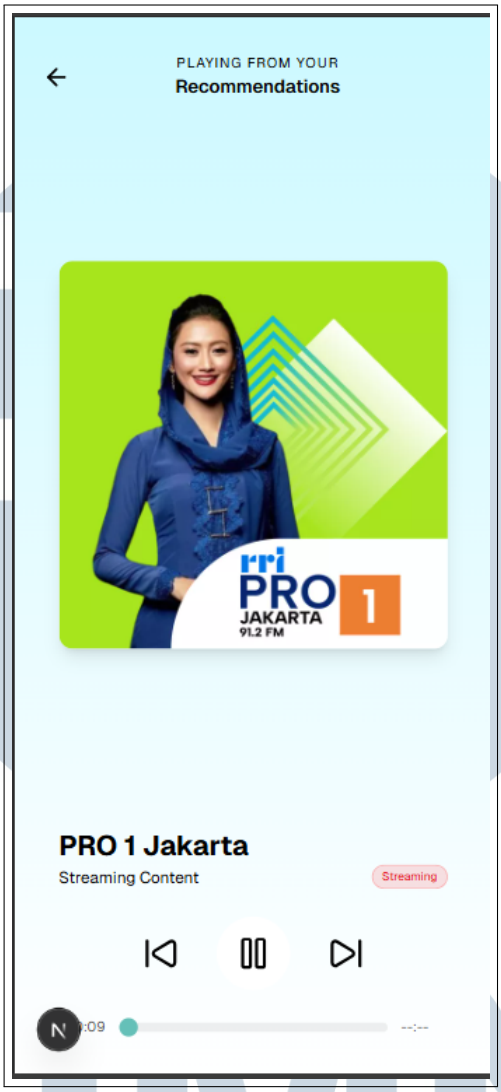
Pada bagian ini juga terlihat proses normalisasi URL streaming. Ketika URL sudah lengkap (misalnya diawali `https://`), maka URL tersebut dapat digunakan langsung. Namun, jika URL masih berupa path/endpoint relatif, URL dibentuk melalui endpoint `/api/proxy` untuk membantu mengatasi kendala CORS dan menstabilkan akses stream. Inisialisasi track dilakukan di dalam `useEffect` pada saat halaman pertama kali dibuka sehingga pemutar langsung siap dimainkan.

Kontrol UI seperti *play/pause* dan *skip back/forward* dipetakan langsung ke fungsi pada context, misalnya `play()`, `pause()`, `next()`, dan `prev()`. Selain itu, progress bar diperbarui berdasarkan `currentTime` dan `duration`. Ketika pengguna melakukan *seek* pada progress, nilai posisi diteruskan ke `seekTo()` sehingga perpindahan posisi pemutaran tetap sinkron dan tidak merusak state di halaman lain.



Gambar 3.41. Hasil implementasi halaman Audio Player Radio pada website (desktop)

Gambar 3.41 memperlihatkan tampilan akhir pemutar radio pada web. Halaman menampilkan identitas stasiun, judul konten yang sedang diputar, indikator status *streaming*, tombol kontrol utama, serta progress bar untuk memantau durasi pemutaran. Tata letak dibuat terpusat agar fokus pengguna berada pada kontrol audio dan informasi yang relevan.



Gambar 3.42. Hasil implementasi Audio Player Radio pada tampilan mobile (responsif)

Gambar 3.42 menunjukkan tampilan pemutar radio pada perangkat mobile. Elemen kontrol diposisikan agar mudah dijangkau, sementara progress bar dibuat tetap terbaca pada lebar layar yang lebih kecil.

3.6 Pengujian Antarmuka dan Evaluasi Pengalaman Pengguna

Pengujian antarmuka pengguna dilakukan untuk memastikan bahwa sistem RRI AI Web dapat berjalan dengan baik secara fungsional serta memberikan pengalaman pengguna yang konsisten di berbagai perangkat. Pengujian difokuskan pada aspek fungsionalitas antarmuka, konsistensi tampilan, dan responsivitas desain.

Pengujian fungsional dilakukan dengan memverifikasi setiap fitur utama, seperti navigasi sidebar, proses pengiriman dan penerimaan pesan pada AI Chat, pemutaran audio melalui Audio Player, serta perpindahan halaman menuju detail konten. Setiap fitur diuji berdasarkan skenario penggunaan umum untuk memastikan tidak terjadi kesalahan tampilan maupun kegagalan interaksi.

Selain itu, dilakukan pengujian responsivitas dengan mengakses aplikasi pada berbagai ukuran layar, mulai dari desktop hingga perangkat mobile. Pengujian ini bertujuan memastikan bahwa tata letak komponen, ukuran teks, serta kontrol interaksi tetap nyaman digunakan meskipun terjadi perubahan resolusi layar.

Evaluasi pengalaman pengguna (UX) dilakukan secara internal melalui diskusi bersama tim UI/UX dan frontend. Masukan yang diperoleh digunakan untuk menyempurnakan alur navigasi, memperjelas hierarki visual, serta meningkatkan kenyamanan interaksi, khususnya pada fitur AI Chat dan pemutar audio. Hasil pengujian dan evaluasi ini menjadi dasar perbaikan pada iterasi pengembangan berikutnya.

Evaluasi ini juga mencakup aspek *usability testing* secara internal, dengan menilai kemudahan penggunaan, kejelasan navigasi, serta kenyamanan interaksi berdasarkan skenario penggunaan umum pengguna layanan audio berbasis web.

3.7 Kendala dan Solusi yang Ditemukan

3.7.1 Kendala

Beberapa kendala yang dihadapi penulis selama pelaksanaan kerja magang pada pengembangan sistem RRI AI adalah sebagai berikut:

- Pola kerja *Work From Home* (WFH) menyebabkan koordinasi pekerjaan tidak selalu berjalan optimal. Pada beberapa kondisi, pembagian tugas antar anggota tim dirasa kurang merata karena komunikasi dilakukan secara daring.
- Perubahan desain antarmuka (UI/UX) yang bersifat dinamis mengharuskan penulis melakukan penyesuaian ulang terhadap komponen frontend yang telah diimplementasikan sebelumnya.
- Kendala teknis pada proses integrasi frontend dengan backend AI, terutama akibat perbedaan struktur data dan perubahan respons API yang memengaruhi proses *rendering* komponen.

3.7.2 Solusi

Untuk mengatasi kendala-kendala tersebut, penulis menerapkan beberapa solusi sebagai berikut:

- Penulis memanfaatkan kegiatan *Daily Stand Up* (DS) sebagai sarana pelaporan progres kerja setiap anggota tim. Melalui DS, pembagian tugas dapat dipantau dengan lebih jelas sehingga pekerjaan dapat didistribusikan secara lebih merata dan terstruktur.
- Selain melalui DS, penulis juga aktif berkomunikasi dengan rekan satu tim melalui *chat* internal untuk mendiskusikan kendala yang dihadapi serta melakukan penyesuaian pembagian tugas apabila diperlukan.
- Penulis menerapkan pengembangan komponen frontend secara modular, sehingga perubahan desain dapat dilakukan dengan lebih fleksibel tanpa harus memodifikasi keseluruhan struktur aplikasi.
- Untuk mengatasi permasalahan integrasi dengan backend, penulis melakukan normalisasi data pada sisi frontend serta berkoordinasi secara rutin dengan tim backend agar struktur data yang digunakan tetap selaras.