

## **BAB V**

### **HASIL DAN PEMBAHASAN**

Bab ini menjelaskan hasil implementasi sistem MoodTune beserta pembahasannya. Sistem telah berhasil dibangun menggunakan framework Next.js untuk frontend dan Tailwind CSS sebagai sistem styling. MoodTune diintegrasikan dengan layanan Spotify menggunakan NextAuth untuk mengakses histori lagu pengguna. Selain itu, proses klasifikasi mood menggunakan data fitur audio disimulasikan dari dataset lokal, sebagai alternatif dari Spotify Audio Features API yang terbatas.

Salah satu fokus utama implementasi adalah membangun antarmuka yang intuitif dan personal agar pengguna dapat berinteraksi dengan sistem secara alami. Berikut ini disajikan hasil implementasi tampilan antarmuka beserta fungsionalitasnya.

#### **5.1 Hasil Implementasi Antarmuka Pengguna**

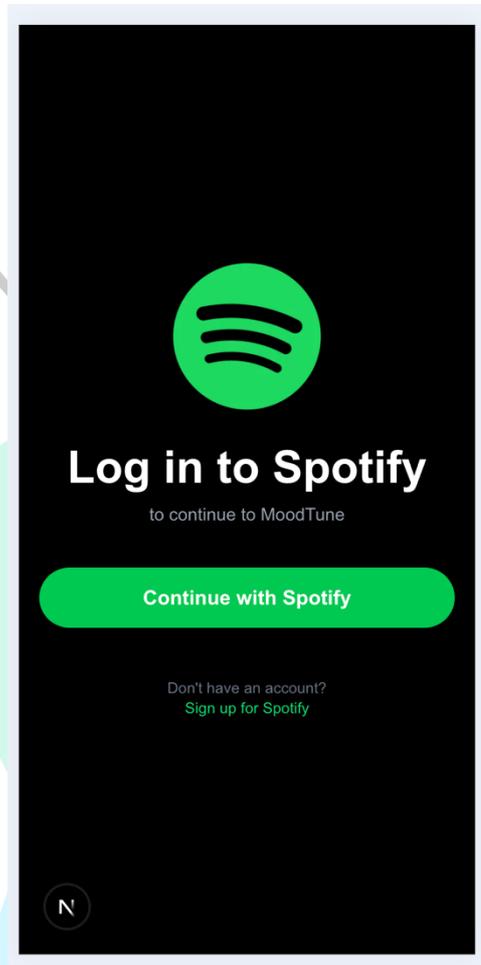
Antarmuka pengguna sistem MoodTune terdiri dari dua halaman utama, yaitu halaman login dan halaman utama (Home). Masing-masing halaman telah berhasil diimplementasikan sesuai dengan kebutuhan sistem. Tampilan utama terdiri dari:

##### **5.1.1 Halaman Login (Autentikasi Spotify)**

Halaman login berfungsi sebagai gerbang awal pengguna untuk mengakses sistem melalui autentikasi Spotify OAuth. Tampilan halaman ini memiliki desain sederhana dengan tata letak terpusat. Komponen yang berhasil ditampilkan meliputi:

- Logo Spotify di bagian atas halaman sebagai penanda layanan yang terintegrasi
- Judul halaman dan tombol “Continue with Spotify” berwarna hijau
- Tampilan latar belakang gelap untuk menciptakan kesan elegan dan fokus

Saat tombol login ditekan, sistem akan mengarahkan pengguna ke halaman otorisasi Spotify, lalu kembali ke halaman utama setelah izin akses diberikan.



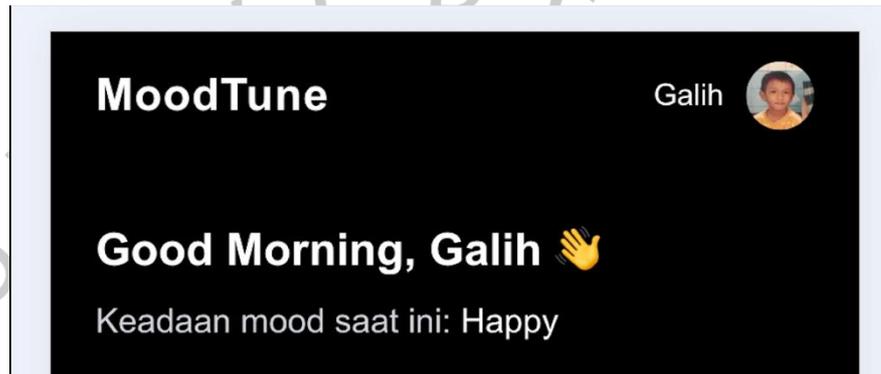
**Gambar 5.1** Halaman Login MoodTune

### **5.1.2 Halaman Utama (Home)**

Setelah proses login berhasil, pengguna akan diarahkan ke halaman utama. Halaman ini memuat beberapa informasi penting terkait kondisi emosional pengguna dan rekomendasi musik yang dihasilkan oleh sistem. Antarmuka halaman utama terdiri dari tiga bagian utama berikut:

### a. Bagian Sapaan dan Prediksi Mood

Pada bagian ini, sistem menampilkan sapaan waktu yang menyesuaikan dengan waktu lokal pengguna, seperti “Selamat pagi” atau “Selamat malam.” Di bawah sapaan tersebut, sistem menampilkan hasil klasifikasi mood pengguna berdasarkan lagu terakhir yang diputar. Output mood seperti *Happy*, *Sad*, *Calm*, atau *Energetic* ditampilkan secara visual untuk memperkuat kesan personal.



Gambar 5.2 Tampilan Sapaan dan Mood Saat Ini

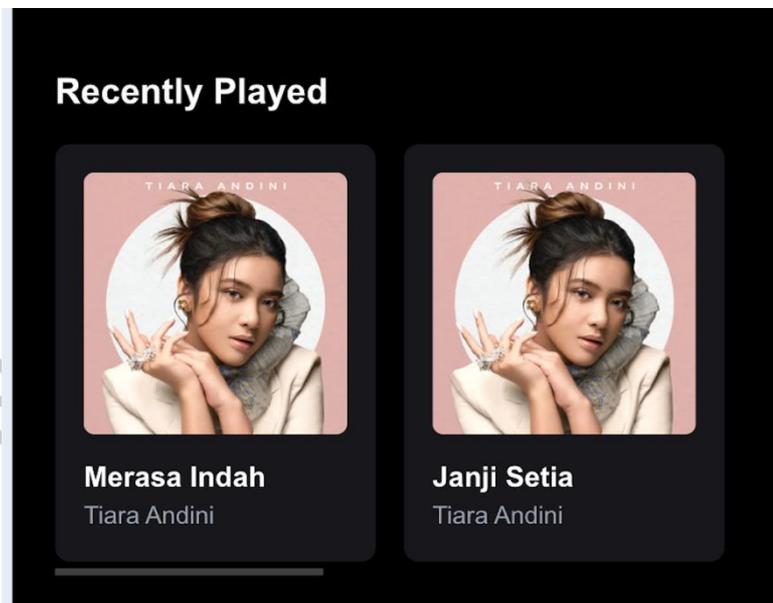
### b. Bagian Histori Lagu (Recently Played)

Bagian ini menampilkan daftar lagu yang terakhir diputar oleh pengguna di Spotify. Data diperoleh melalui endpoint `/me/player/recently-played`, yang mengembalikan respons dalam format JSON. Beberapa field penting dari data respons yang digunakan dalam sistem ini antara lain:

- `track.name`: judul lagu
- `track.artists[].name`: nama penyanyi atau grup musik
- `track.album.images[0].url`: URL sampul album (cover)
- `played_at`: waktu pemutaran lagu
- `track.id`: ID lagu yang digunakan sebagai referensi internal dan pemanggilan data lanjutan

Informasi dari field-field tersebut ditampilkan dalam bentuk kartu horizontal pada antarmuka, yang memuat sampul album, judul lagu, dan nama artis. Selain sebagai

tampilan histori, data ini juga digunakan sebagai dasar input untuk proses klasifikasi mood dalam sistem.

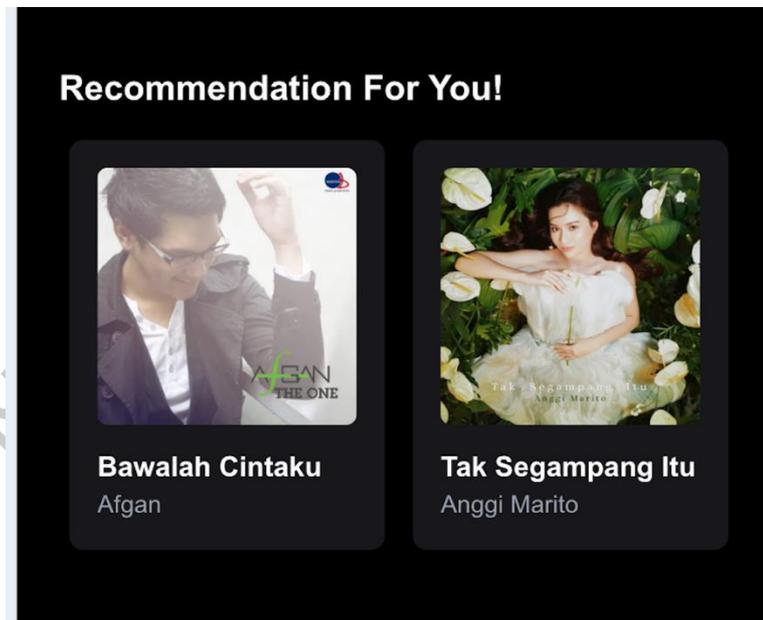


Gambar 5.3 Tampilan Daftar Lagu Terakhir

### c. Bagian Rekomendasi Lagu

Setelah mood pengguna berhasil diidentifikasi, sistem menampilkan daftar lagu yang direkomendasikan berdasarkan hasil klasifikasi tersebut. Lagu-lagu ini diambil dari dataset lokal yang telah diberi label mood, dan disesuaikan agar mencerminkan atau memperkuat suasana hati pengguna saat itu. Daftar

rekomendasi disusun dalam format grid responsif, agar mudah diakses dan dinavigasi oleh pengguna.



Gambar 5.4 Tampilan Daftar Rekomendasi Lagu

## 5.2 Hasil Implementasi Model dan Alur Sistem

Sistem MoodTune telah berhasil dibangun dan dijalankan secara parsial, mencakup tiga komponen utama, yaitu integrasi Spotify API, model klasifikasi berbasis Deep Neural Network (DNN), dan antarmuka pengguna (frontend). Berikut adalah hasil implementasi dari masing-masing komponen:

### 1. Spotify API

Spotify API digunakan untuk mengambil riwayat lagu yang terakhir diputarkan oleh pengguna melalui endpoint `/me/player/recently-played`. Data yang diambil dari respons tersebut mencakup beberapa field penting seperti:

- `track.name` (judul lagu)
- `track.artists[].name` (nama artis)
- `track.album.images[0].url` (URL sampul album)

- played\_at (waktu pemutaran)
- track.id (ID lagu untuk referensi lebih lanjut)

Namun, fitur audio seperti valence, energy, tempo, dan danceability tidak tersedia karena keterbatasan akses API. Oleh karena itu, sistem mensimulasikan data tersebut menggunakan dataset lokal dengan struktur serupa untuk mendukung proses klasifikasi mood.

## 2. Model Deep Neural Network

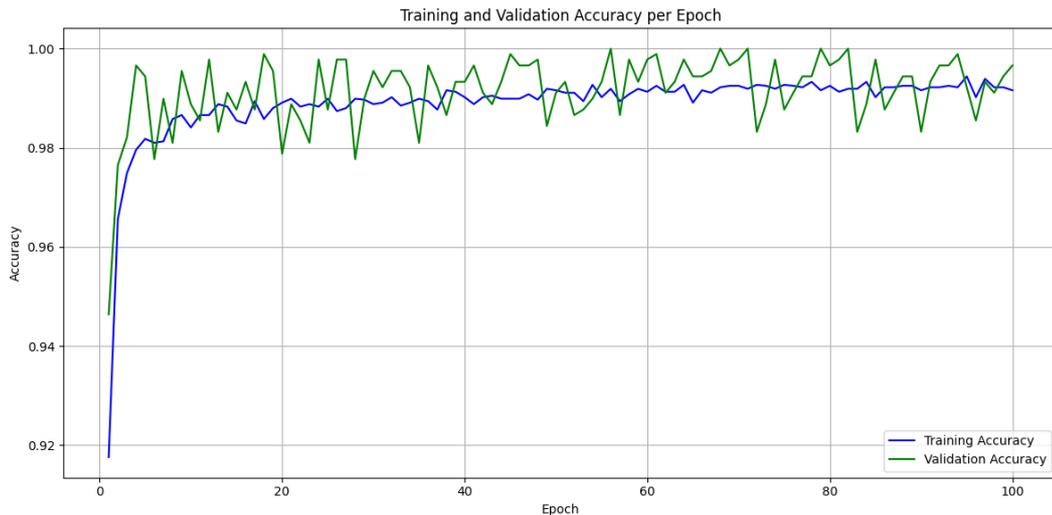
Model klasifikasi berbasis Deep Neural Network (DNN) telah dikembangkan untuk memprediksi suasana hati pengguna berdasarkan fitur numerik dari data audio. Dengan struktur jaringan berlapis dan aktivasi non-linear, model ini dirancang untuk menangkap hubungan kompleks antar fitur.

**Konfigurasi arsitektur model** meliputi:

- **Input Layer:** menerima empat fitur numerik, yaitu *valence*, *energy*, *tempo*, dan *danceability*.
- **Hidden Layer:** terdiri dari dua lapisan tersembunyi dengan masing-masing 32 neuron, menggunakan fungsi aktivasi ReLU.
- **Output Layer:** empat neuron mewakili kelas mood (Happy, Sad, Calm, Energetic), dengan fungsi aktivasi Softmax untuk menghasilkan probabilitas klasifikasi.
- **Parameter pelatihan:** model dilatih selama 100 epoch dengan *batch size* sebesar 8 menggunakan optimizer Adam.

Proses pelatihan dilakukan menggunakan data hasil pseudo-labeling dari pendekatan rule-based, dengan alokasi 20% data untuk validasi. Hasil evaluasi menunjukkan bahwa model mampu mencapai **akurasi pelatihan sebesar 99,16%** dan **akurasi validasi sebesar 99,44%** pada epoch ke-100.

Grafik di bawah memperlihatkan tren akurasi selama proses pelatihan:



Dari grafik tersebut, terlihat bahwa model mengalami peningkatan performa yang konsisten pada awal pelatihan dan kemudian mencapai stabilitas. Perbedaan antara akurasi pelatihan dan validasi relatif kecil, menunjukkan bahwa model tidak mengalami overfitting secara signifikan, serta mampu melakukan generalisasi terhadap data baru dengan baik.

Dengan hasil tersebut, model DNN yang dibangun dapat diandalkan sebagai komponen inti dalam sistem rekomendasi musik berbasis mood, terutama ketika diintegrasikan dengan data real-time dari Spotify.

### 3. Antarmuka Pengguna (Frontend)

Tampilan antarmuka pengguna berhasil diimplementasikan menggunakan framework Next.js dan Tailwind CSS. Pengguna dapat masuk melalui autentikasi Spotify, melihat prediksi mood berdasarkan lagu terakhir, melihat riwayat lagu yang diputar, serta menerima rekomendasi lagu yang sesuai dengan hasil klasifikasi mood. Tampilan antarmuka dibangun secara modular agar mudah dikembangkan di masa depan.

### 5.3 Hasil Implementasi Interaksi Sistem

Untuk menggambarkan alur proses dari awal hingga akhir, sistem MoodTune dirancang berdasarkan beberapa skenario interaksi yang kemudian diimplementasikan ke dalam sistem. Adapun alur yang berhasil diterapkan meliputi:

- **Login Spotify**  
Pengguna diarahkan ke halaman login untuk melakukan autentikasi melalui mekanisme OAuth yang disediakan oleh Spotify.
- **Pengambilan Lagu Terakhir**  
Sistem menggunakan Spotify API untuk mengambil riwayat lagu yang baru diputar oleh pengguna. Data ini digunakan sebagai input awal dalam sistem.
- **Klasifikasi Mood (Simulasi)**  
Data lagu yang telah diambil digunakan sebagai referensi untuk melakukan simulasi klasifikasi suasana hati menggunakan model DNN yang telah dibangun.
- **Rekomendasi Lagu**  
Berdasarkan hasil klasifikasi mood, sistem menampilkan daftar lagu rekomendasi dari dataset lokal yang disesuaikan dengan mood pengguna.

Seluruh alur interaksi ini telah divisualisasikan dalam bentuk use case diagram, activity diagram, dan sequence diagram, serta berhasil diimplementasikan sesuai skenario yang dirancang.

### 5.4 Pembahasan

Berdasarkan hasil implementasi dan simulasi yang telah dilakukan, sistem MoodTune menunjukkan potensi besar sebagai sistem rekomendasi musik berbasis suasana hati. Adapun beberapa hal penting yang dapat dibahas dari hasil pengembangan adalah sebagai berikut:

- **Ketergantungan terhadap Spotify API**

Spotify API merupakan bagian penting dalam sistem ini, terutama dalam pengambilan data lagu pengguna. Namun, keterbatasan akses pada endpoint /audio-features menjadi tantangan utama dalam integrasi penuh. Sebagai solusi sementara, fitur audio disimulasikan menggunakan dataset lokal dengan struktur serupa.

- **Model DNN sebagai Solusi Klasifikasi Mood**

Penggunaan DNN memberikan kemampuan klasifikasi yang baik terhadap pola-pola kompleks dalam data musik. Meskipun belum diujikan secara penuh dalam sistem berjalan, arsitektur model telah disiapkan dan siap diintegrasikan jika akses ke data real tersedia.

- **Antarmuka Modular dan Mudah Diperluas**

Desain antarmuka yang modular membuat pengembangan sistem menjadi fleksibel. Komponen seperti prediksi mood, histori lagu, dan rekomendasi dapat dikembangkan lebih lanjut tanpa perlu mengubah struktur sistem secara keseluruhan.

- **Rencana Pengujian Telah Disusun**

Pengujian terhadap sistem belum dilakukan secara aktual, namun telah dirancang dengan metode black-box untuk fungsi antarmuka serta white-box untuk logika sistem internal. Pengujian ini dapat segera dilaksanakan apabila implementasi penuh telah tersedia.

Secara keseluruhan, implementasi sistem MoodTune telah mencakup seluruh komponen utama, mulai dari pengambilan data pengguna, klasifikasi mood, hingga penyajian rekomendasi lagu. Meskipun masih dalam tahap simulasi, sistem telah menunjukkan kesiapan untuk diintegrasikan secara penuh dalam versi produksi apabila hambatan teknis seperti akses API dapat diselesaikan.

## **BAB VI**

### **PENUTUPAN**

Bab ini berisi rangkuman hasil penelitian serta rekomendasi pengembangan di masa mendatang. Penelitian ini berfokus pada perancangan sistem rekomendasi musik berbasis suasana hati (mood) dengan dukungan Spotify API dan pendekatan klasifikasi menggunakan Deep Neural Network (DNN).

#### **6.1 Kesimpulan**

Berdasarkan hasil perancangan dan analisis yang telah dilakukan, dapat disimpulkan beberapa poin utama sebagai berikut:

1. MoodTune dirancang sebagai aplikasi web yang bertujuan memberikan rekomendasi lagu berdasarkan suasana hati pengguna, yang diprediksi melalui lagu terakhir yang diputar.
2. Sistem mengintegrasikan Spotify API untuk memperoleh data pengguna serta histori lagu, dan menggunakan fitur-fitur audio seperti valence, energy, dan danceability sebagai dasar input klasifikasi.
3. Model klasifikasi menggunakan Deep Neural Network (DNN) dengan tiga lapisan tersembunyi, aktivasi ReLU, dropout untuk regularisasi, serta output dengan fungsi aktivasi softmax.
4. Antarmuka pengguna dikembangkan menggunakan Next.js dan Tailwind CSS, dengan desain responsif dan terstruktur untuk memudahkan interaksi pengguna.
5. Karena adanya pembatasan akses dari Spotify, sistem belum dapat diimplementasikan sepenuhnya. Proses klasifikasi dilakukan secara simulatif menggunakan data lokal yang menyerupai struktur Spotify.
6. Perancangan sistem telah mencakup dokumentasi menyeluruh, meliputi use case diagram, activity diagram, sequence diagram, serta strategi pengujian black-box dan white-box untuk mendukung implementasi ke tahap berikutnya.

## 6.2 Saran

Sebagai bentuk pengembangan lanjutan, berikut beberapa saran yang dapat dipertimbangkan:

1. Pengajuan akses resmi ke Spotify API (khususnya endpoint audio-features) sebaiknya dilakukan untuk memungkinkan sistem berjalan secara real-time dan aktual.
2. Perlu dieksplorasi kemungkinan menggunakan sumber data alternatif, seperti analisis lirik, metadata sosial, atau preferensi eksplisit pengguna, guna melengkapi fitur audio dalam klasifikasi mood.
3. Sistem dapat dikembangkan lebih lanjut untuk mendukung rekomendasi yang lebih personal, misalnya berdasarkan histori mood pengguna atau waktu tertentu (pagi, malam, akhir pekan, dsb.).
4. Model DNN yang digunakan dapat dioptimalkan lebih lanjut melalui tuning hyperparameter, eksplorasi teknik data augmentation, atau penerapan arsitektur alternatif seperti CNN atau LSTM.
5. Evaluasi sistem melalui user testing langsung sangat disarankan untuk mengukur kualitas rekomendasi dan kepuasan pengguna secara empiris.