

# Laporan Evaluasi Desain Perangkat Lunak "Musik Jamak": Analisis Fungsionalitas dan Usability Lintas Platform

## Bagian 1: Desain Konseptual dan Arsitektur Sistem "Musik Jamak"

### 1.1. Tinjauan Umum Proyek dan Metodologi Evaluasi

Proyek "Musik Jamak" merupakan sebuah inisiatif perancangan perangkat lunak yang bertujuan untuk mendesain dan memodelkan sebuah aplikasi layanan *streaming* musik komprehensif. Proyek ini secara konseptual mereplikasi fungsionalitas inti dan pengalaman pengguna yang ditawarkan oleh pemimpin pasar, Spotify. Sebagaimana diamanatkan oleh tugas Ujian Tengah Semester mata kuliah Evaluasi Desain Perangkat Lunak, laporan ini menyajikan evaluasi mendalam terhadap desain perancangan tersebut.<sup>1</sup> Fokus utama evaluasi ini adalah untuk menganalisis secara kritis dua aspek fundamental dari kualitas perangkat lunak: **Fungsionalitas** (*Functionality*) dan **Usability** (Kegunaan).

Tujuan dari laporan evaluasi ini adalah untuk memberikan penilaian yang objektif dan terstruktur terhadap kualitas desain "Musik Jamak" sebelum tahap implementasi. Analisis ini mencakup tiga platform target yang menjadi standar industri untuk aplikasi sejenis, yaitu **Desktop**, **Mobile**, dan **Web**. Dengan mengevaluasi desain pada ketiga platform ini, laporan ini akan mengidentifikasi kekuatan, kelemahan, serta potensi inkonsistensi dalam pengalaman pengguna lintas perangkat.

Untuk mencapai tujuan tersebut, laporan ini mengadopsi metodologi evaluasi multi-aspek yang menggabungkan pendekatan formal dan heuristik. Metodologi ini terdiri dari tiga pilar utama:

1. **Pemodelan Sistem menggunakan Unified Modeling Language (UML):** Sebelum melakukan evaluasi, arsitektur dan alur kerja sistem "Musik Jamak" akan divisualisasikan menggunakan diagram UML standar. Ini mencakup *Use Case Diagram* untuk mendefinisikan lingkup fungsional, *Class Diagram* untuk memodelkan struktur data statis, dan *Sequence Diagram* untuk menggambarkan interaksi dinamis antar komponen. Pemodelan ini memberikan dasar yang konkret dan tidak ambigu untuk analisis selanjutnya.
2. **Evaluasi Fungsionalitas Berbasis Standar ISO/IEC 25010:** Untuk memastikan penilaian fungsionalitas yang rigor dan objektif, evaluasi akan mengacu pada kerangka kerja standar internasional ISO/IEC 25010, khususnya pada karakteristik *Functional Suitability* dan sub-karakteristiknya.<sup>2</sup> Pendekatan ini memungkinkan analisis sistematis terhadap kelengkapan, kebenaran, dan kesesuaian fitur yang dirancang.
3. **Evaluasi Usability Berbasis 10 Heuristik Nielsen:** Aspek kegunaan antarmuka akan dievaluasi menggunakan 10 Heuristik Usability yang dikembangkan oleh Jakob Nielsen. Kerangka kerja ini, yang telah menjadi standar de facto dalam industri desain UX/UI, menyediakan serangkaian prinsip untuk mengidentifikasi potensi masalah usability dalam desain antarmuka pengguna.<sup>3</sup>

Melalui penerapan metodologi yang komprehensif ini, laporan ini bertujuan untuk menghasilkan analisis yang tidak hanya memenuhi persyaratan akademis, tetapi juga memberikan wawasan strategis yang berharga mengenai desain aplikasi *streaming* musik modern.

## 1.2. Pemodelan Sistem dengan Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan visual standar yang esensial dalam rekayasa perangkat lunak. Penggunaan UML memungkinkan para pemangku kepentingan—mulai dari analis, desainer, hingga pengembang—untuk memvisualisasikan, merinci, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Dalam konteks evaluasi "Musik Jamak", diagram-diagram UML berfungsi sebagai fondasi analitis. Diagram-diagram ini menerjemahkan konsep abstrak aplikasi menjadi model terstruktur yang dapat dianalisis secara sistematis, baik dari segi fungsionalitas yang ditawarkan maupun dari alur interaksi yang akan dialami oleh pengguna.

### 1.2.1. Diagram Use Case (Use Case Diagram)

Diagram Use Case berfungsi untuk memetakan fungsionalitas sistem dari perspektif pengguna atau "aktor". Diagram ini secara efektif mendefinisikan batasan sistem (*system boundary*), mengidentifikasi siapa saja yang berinteraksi dengannya, dan fungsionalitas utama apa yang dapat mereka lakukan.<sup>5</sup> Untuk "Musik Jamak", diagram ini mengklarifikasi lingkup fitur yang akan dievaluasi.

Aktor Utama:

Berdasarkan analisis fungsionalitas layanan streaming musik modern, tiga aktor utama diidentifikasi untuk sistem "Musik Jamak" <sup>6</sup>:

- **Pengguna (User):** Merupakan aktor primer yang menjadi target utama aplikasi. Aktor ini dapat dibagi lagi menjadi pengguna gratis dan pengguna premium, yang berinteraksi dengan sebagian besar fitur seperti mendengarkan musik, membuat *playlist*, dan mengikuti artis.
- **Artis (Artist):** Aktor ini merepresentasikan kreator konten. Fungsionalitas utama mereka adalah mengunggah musik, mengelola album, dan melihat analitik performa karya mereka.<sup>6</sup>
- **Administrator Sistem (Admin):** Aktor internal yang bertanggung jawab atas pemeliharaan sistem. Tugas mereka mencakup manajemen akun pengguna, pengelolaan langganan dan pembayaran, serta memastikan integritas dan ketersediaan konten di platform.<sup>6</sup>

Use Case Utama:

Use case utama sistem "Musik Jamak" mencakup serangkaian interaksi fundamental yang memungkinkan berjalannya layanan. Disintesis dari berbagai sumber, use case ini meliputi 8:

- **Manajemen Akun:** Mencakup *Registrasi*, *Login*, dan *Mengelola Profil*.
- **Penemuan Konten:** Mencakup *Mencari Musik* (berdasarkan lagu, artis, album), *Menelusuri Rekomendasi*, dan *Melihat Detail Konten*.
- **Interaksi Konten:** Mencakup *Memutar Musik*, *Menyimpan ke Perpustakaan*, dan *Menyukai Lagu*.
- **Manajemen Playlist:** Mencakup *Membuat Playlist*, *Menambah/Menghapus Lagu dari Playlist*, dan *Berbagi Playlist*.
- **Fitur Sosial:** Mencakup *Mengikuti Pengguna/Artis* dan *Melihat Aktivitas Teman*.
- **Fungsi Spesifik Artis:** Mencakup *Mengunggah Musik* dan *Mengelola Profil Artis*.
- **Fungsi Spesifik Admin:** Mencakup *Mengelola Pengguna* dan *Mengelola Langganan*.

Berikut adalah representasi visual dari Diagram Use Case untuk sistem "Musik Jamak":

## 1.2.2. Diagram Kelas (Class Diagram)

Diagram Kelas menyediakan pandangan statis terhadap arsitektur sistem. Ini berfungsi sebagai cetak biru yang mendefinisikan kelas-kelas utama, atribut (data) yang dimilikinya, metode (operasi) yang dapat dilakukannya, serta hubungan struktural antar kelas tersebut.<sup>10</sup> Diagram ini krusial untuk memahami bagaimana data diorganisir dan bagaimana entitas-entitas sistem saling terkait, yang secara langsung mempengaruhi implementasi fungsionalitas.

Kelas Inti, Atribut, dan Metode:

Kelas-kelas inti berikut ini disintesis dari analisis model data untuk aplikasi serupa dan merupakan representasi fundamental dari domain "Musik Jamak" 10:

- **User:** Merepresentasikan pengguna aplikasi.
  - **Atribut:** userID: int, username: string, passwordHash: string, email: string, subscriptionType: enum.
  - **Metode:** register(), login(), updateProfile(), createPlaylist(), followArtist(artist: Artist).
- **Song:** Merepresentasikan satu entitas lagu.
  - **Atribut:** songID: int, title: string, duration: int, genre: string, releaseDate: date, filePath: string.
  - **Metode:** play(), getLyrics(), getDetails().
- **Artist:** Merepresentasikan musisi atau kreator.
  - **Atribut:** artistID: int, name: string, biography: string, genre: string.
  - **Metode:** uploadAlbum(album: Album), viewAnalytics().
- **Album:** Merepresentasikan koleksi lagu.
  - **Atribut:** albumID: int, title: string, releaseDate: date, coverArtURL: string.
  - **Metode:** listSongs(), getAlbumDetails().
- **Playlist:** Merepresentasikan daftar putar yang dibuat pengguna.
  - **Atribut:** playlistID: int, name: string, description: string, creationDate: date, isPublic: bool.
  - **Metode:** addSong(song: Song), removeSong(song: Song), share(), setCollaborative(status: bool).

Hubungan Antar Kelas:

Hubungan antar kelas mendefinisikan bagaimana objek-objek berinteraksi dan saling bergantung:

- **User dan Playlist:** Satu User dapat memiliki (membuat) banyak Playlist. Ini adalah hubungan *one-to-many*.
- **Playlist dan Song:** Satu Playlist dapat berisi banyak Song, dan satu Song dapat dimasukkan ke dalam banyak Playlist. Ini adalah hubungan *many-to-many*, yang dalam implementasi basis data relasional biasanya memerlukan tabel perantara (misalnya,

Playlist\_Songs).

- **Artist dan Album:** Satu Artist dapat merilis banyak Album. Ini adalah hubungan *one-to-many*.
- **Album dan Song:** Satu Album terdiri dari banyak Song. Ini adalah hubungan komposisi (*composition*), sebuah bentuk agregasi yang kuat, di mana Song tidak dapat eksis tanpa Album. Hubungan ini adalah *one-to-many*.

### 1.2.3. Diagram Sekuens (Sequence Diagram)

Berbeda dengan pandangan statis dari Diagram Kelas, Diagram Sekuens memodelkan aspek dinamis sistem. Diagram ini menggambarkan bagaimana serangkaian objek berinteraksi satu sama lain dari waktu ke waktu untuk menyelesaikan sebuah *use case* atau skenario tertentu.<sup>12</sup> Diagram ini sangat berguna untuk memvalidasi alur logika dan mengidentifikasi potensi hambatan (*bottlenecks*) dalam komunikasi antar komponen sistem.

Skenario Kritis: "Memutar Lagu (Play Song)"

Memutar lagu adalah salah satu *use case* paling fundamental dan sering terjadi. Menganalisis urutan interaksinya memberikan wawasan tentang arsitektur real-time aplikasi. Alur berikut disintesis dari berbagai sumber yang menjelaskan arsitektur layanan streaming 13:

1. **Aksi Pengguna:** User berinteraksi dengan UserInterface dengan memilih sebuah lagu. Ini mengirimkan pesan *selectSong(songID)*.
2. **Permintaan Metadata:** UserInterface kemudian meminta detail lagu (judul, artis, sampul album) dari SongMetadataService dengan pesan *getSongDetails(songID)*. Layanan ini bertanggung jawab atas semua informasi deskriptif tentang konten.
3. **Akses Basis Data:** SongMetadataService melakukan kueri ke Database untuk mengambil data yang diminta dan mengembalikannya ke UserInterface.
4. **Pembaruan Antarmuka:** UserInterface menampilkan informasi lagu kepada pengguna dan, secara bersamaan, mengirimkan permintaan pemutaran ke MediaPlayer dengan pesan *play(songID)*.
5. **Permintaan Stream:** MediaPlayer (komponen klien yang menangani audio) meminta aliran data audio dari SongStreamingService menggunakan *songID* sebagai referensi.
6. **Pengambilan File Audio:** SongStreamingService (layanan *backend* yang menangani pengiriman file audio) mengambil file audio aktual dari sistem penyimpanan objek, seperti Amazon S3 atau HDFS, yang dioptimalkan untuk pengiriman file besar.<sup>15</sup>
7. **Streaming Audio:** Layanan tersebut mulai mengirimkan data audio dalam bentuk *stream* (potongan-potongan data) kembali ke MediaPlayer.
8. **Pemutaran:** MediaPlayer menerima *stream*, melakukan *buffering*, dan memulai pemutaran audio melalui perangkat keluaran suara.

### 1.3. Filosofi Desain Antarmuka Pengguna (UI/UX)

Filosofi desain antarmuka "Musik Jamak" secara sadar mengadopsi prinsip-prinsip yang telah terbukti berhasil pada Spotify. Desain ini tidak hanya berfokus pada estetika, tetapi juga pada penciptaan pengalaman pengguna yang intuitif, praktis, dan menarik, terutama bagi target demografis utama, yaitu generasi muda dan dewasa awal yang merupakan penikmat musik aktif.<sup>16</sup>

#### Prinsip-Prinsip Utama Desain:

- **Desain Minimalis dan Estetis (Aesthetic and Minimalist Design):** Prinsip ini diwujudkan melalui penggunaan antarmuka yang bersih dan tidak berantakan. Tema gelap (*dark theme*) menjadi pilihan utama, yang berfungsi ganda: secara estetis menonjolkan sampul album yang berwarna-warni dan secara fungsional mengurangi ketegangan mata di lingkungan dengan cahaya rendah.<sup>17</sup> Desain minimalis memastikan bahwa fokus pengguna tetap pada konten utama, yaitu musik.
- **Tata Letak Hierarkis (Hierarchical Grid Layout):** Konten disajikan dalam sebuah grid yang terstruktur secara hierarkis. Elemen-elemen yang paling penting atau sering digunakan, seperti kolom pencarian, ditempatkan di posisi yang paling menonjol (misalnya, bagian atas layar). Konten yang dipersonalisasi seperti *playlist* "Recently Played" atau "Daily Mix" diberikan ruang visual yang lebih besar untuk menarik perhatian pengguna dan mendorong interaksi.<sup>16</sup>
- **Konsistensi dan Standar (Consistency and Standards):** Untuk mengurangi beban kognitif pengguna, "Musik Jamak" menerapkan konsistensi yang ketat di seluruh platform. Ikonografi (misalnya, simbol segitiga untuk 'Play', dua garis vertikal untuk 'Pause', panah bersilangan untuk 'Shuffle') menggunakan standar universal yang sudah dikenal pengguna dari aplikasi media lainnya. Tata letak menu navigasi utama (Beranda, Cari, Koleksi) juga dipertahankan di lokasi yang sama di berbagai layar untuk menciptakan pengalaman yang dapat diprediksi dan mudah dipelajari.<sup>16</sup>
- **Tipografi yang Jelas dan Mudah Dibaca:** Pemilihan tipografi adalah elemen krusial dalam desain UI. "Musik Jamak" mengadopsi jenis *font sans-serif* yang modern dan bersih, seperti Gotham yang digunakan oleh Spotify. Karakteristik *font* yang tebal dan tegak memastikan keterbacaan (*readability*) yang tinggi, memungkinkan pengguna untuk dengan cepat memindai judul lagu, nama artis, dan item menu, bahkan pada layar perangkat *mobile* yang lebih kecil.<sup>16</sup>
- **Penggunaan Warna sebagai Identitas Merek:** Meskipun didominasi oleh tema gelap, penggunaan warna aksen yang cerah dan khas (misalnya, hijau) pada elemen-elemen interaktif seperti tombol, *slider*, dan tautan aktif berfungsi sebagai penanda visual (*visual cue*) dan sekaligus memperkuat identitas merek aplikasi.<sup>16</sup>

Penerapan filosofi desain ini lebih dari sekadar keputusan gaya; ini adalah strategi fungsional yang mendalam. Tema gelap, misalnya, tidak hanya terlihat modern tetapi juga memiliki manfaat teknis yang nyata pada perangkat *mobile* dengan layar OLED, di mana piksel hitam tidak mengonsumsi daya, sehingga dapat menghemat masa pakai baterai. Tata letak hierarkis secara langsung mendukung tujuan bisnis inti dari platform *streaming*: meningkatkan *engagement* dan retensi pengguna. Dengan menempatkan konten yang dipersonalisasi secara algoritmik (seperti "Discover Weekly" atau "Daily Mix") di posisi yang paling menonjol, desain antarmuka secara aktif mendorong pengguna untuk menemukan musik baru yang mereka sukai, yang pada akhirnya meningkatkan nilai platform bagi mereka dan membuat mereka terus kembali.<sup>19</sup> Oleh karena itu, evaluasi *usability* terhadap "Musik Jamak" harus melampaui penilaian subjektif tentang "apakah tampilannya bagus," dan lebih fokus pada "apakah setiap pilihan desain secara efektif mendukung tujuan pengguna dan sasaran strategis aplikasi."

## Bagian 2: Analisis Evaluasi Fungsionalitas

### 2.1. Kerangka Evaluasi: Karakteristik Functional Suitability (ISO/IEC 25010)

Untuk memastikan analisis fungsionalitas "Musik Jamak" dilakukan dengan cara yang sistematis, objektif, dan sesuai dengan standar industri, laporan ini mengadopsi kerangka kerja yang didefinisikan dalam ISO/IEC 25010. Standar ini menyediakan model kualitas produk perangkat lunak yang komprehensif dan diakui secara internasional, memungkinkan evaluasi yang terstruktur dan dapat dipertanggungjawabkan.<sup>2</sup>

Dari delapan karakteristik kualitas yang didefinisikan dalam standar tersebut, evaluasi ini akan berfokus secara spesifik pada **Functional Suitability** (Kesesuaian Fungsional). Karakteristik ini didefinisikan sebagai "sejauh mana suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan dalam kondisi tertentu".<sup>20</sup> Dengan kata lain, karakteristik ini mengukur apakah perangkat lunak benar-benar melakukan apa yang seharusnya dilakukannya untuk memenuhi kebutuhan pengguna.

Evaluasi *Functional Suitability* akan dipecah lebih lanjut ke dalam tiga sub-karakteristiknya, yang memungkinkan analisis yang lebih granular dan mendalam<sup>20</sup>:

1. **Kelengkapan Fungsional (Functional Completeness):** Menilai apakah serangkaian fitur yang disediakan oleh "Musik Jamak" telah mencakup semua tugas yang dibutuhkan dan tujuan yang ingin dicapai oleh pengguna target. Pertanyaan kuncinya adalah: "Apakah ada fungsionalitas penting yang hilang?"
2. **Kebenaran Fungsional (Functional Correctness):** Menilai apakah perangkat lunak memberikan hasil yang benar dan akurat sesuai dengan spesifikasinya. Pertanyaan kuncinya adalah: "Apakah fitur-fitur yang ada bekerja dengan benar dan memberikan output yang diharapkan?"
3. **Kesesuaian Fungsional (Functional Appropriateness):** Menilai sejauh mana fungsi-fungsi yang ada benar-benar membantu dan memfasilitasi pengguna dalam menyelesaikan tugas-tugas mereka secara efisien. Pertanyaan kuncinya adalah: "Apakah fitur-fitur ini dirancang dengan cara yang mendukung alur kerja pengguna secara efektif?"

Dengan menggunakan kerangka kerja ini, analisis fungsionalitas dapat bergerak melampaui sekadar inventarisasi fitur, menuju evaluasi kualitatif tentang seberapa baik fitur-fitur tersebut benar-benar melayani tujuannya.

## 2.2. Evaluasi Fitur Inti dan Pendukung "Musik Jamak"

Analisis berikut mengevaluasi serangkaian fitur "Musik Jamak", yang didasarkan pada fitur-fitur standar industri yang ada di Spotify, terhadap tiga sub-karakteristik *Functional Suitability*.

Fitur Inti (Core Features):

Fitur-fitur ini merupakan fondasi dari aplikasi streaming musik dan harus berfungsi tanpa cela.

- **Streaming Audio:** Kemampuan untuk memutar audio secara *real-time*.
  - *Completeness:* Lengkap. Desain mencakup pemutaran audio berkualitas tinggi (hingga 320 kbps untuk akun premium), *gapless playback* untuk transisi lagu yang mulus, dan *adaptive bitrate streaming* yang menyesuaikan kualitas audio dengan kecepatan koneksi internet pengguna untuk meminimalkan *buffering*.<sup>17</sup>
  - *Correctness:* Asumsi: Bekerja sesuai harapan. Lagu diputar dengan kualitas yang dipilih dan transisi antar lagu berjalan mulus.
  - *Appropriateness:* Sangat sesuai. Ini adalah fungsi paling fundamental dari aplikasi.
- **Pencarian Lanjutan:** Mekanisme untuk menemukan konten.
  - *Completeness:* Lengkap. Desain mendukung pencarian tidak hanya berdasarkan judul lagu, artis, dan album, tetapi juga berdasarkan genre, *playlist*, dan bahkan

cuplikan lirik, yang secara signifikan memperluas kemampuan penemuan konten.<sup>16</sup>

- **Correctness:** Asumsi: Hasil pencarian relevan dan akurat. Pencarian lirik berhasil mencocokkan lagu yang benar.
- **Appropriateness:** Sangat sesuai. Fitur pencarian yang kuat adalah gerbang utama bagi pengguna untuk mengakses perpustakaan musik yang luas.
- **Manajemen Playlist:** Kemampuan untuk mengkurasikan koleksi lagu pribadi.
  - **Completeness:** Lengkap. Pengguna dapat membuat, mengedit (mengubah nama, deskripsi, urutan lagu), menghapus, dan membagikan *playlist*. Fitur *playlist* kolaboratif, di mana beberapa pengguna dapat menambahkan lagu ke satu *playlist*, juga termasuk dalam desain.<sup>19</sup>
  - **Correctness:** Asumsi: Semua operasi CRUD (*Create, Read, Update, Delete*) pada *playlist* dan lagu di dalamnya berfungsi dengan benar dan tersinkronisasi di seluruh perangkat pengguna.
  - **Appropriateness:** Sangat sesuai. Personalisasi melalui *playlist* adalah inti dari pengalaman *streaming* musik modern.
- **Mode Offline:** Kemampuan mendengarkan tanpa koneksi internet.
  - **Completeness:** Lengkap (untuk pengguna premium). Desain memungkinkan pengguna mengunduh *playlist*, album, dan podcast ke perangkat mereka untuk pemutaran *offline*.<sup>17</sup>
  - **Correctness:** Asumsi: Proses pengunduhan berjalan lancar dengan indikator kemajuan yang jelas, dan konten yang diunduh dapat diputar dengan sempurna saat perangkat tidak terhubung ke internet.
  - **Appropriateness:** Sangat sesuai dan krusial. Fitur ini menjawab kebutuhan utama pengguna *mobile* yang sering berada di area dengan koneksi terbatas (misalnya, di kereta bawah tanah, pesawat) atau bagi mereka yang ingin menghemat kuota data seluler.

#### Fitur Personalisasi dan Penemuan (Discovery Features):

Fitur-fitur ini adalah pembeda utama yang menciptakan pengalaman unik bagi setiap pengguna dan mendorong retensi.

- **Rekomendasi Berbasis AI:** Algoritma yang menyajikan musik baru kepada pengguna.
  - **Completeness:** Lengkap. Desain mencakup beberapa format rekomendasi yang telah terbukti berhasil: *playlist* mingguan ("Discover Weekly"), *playlist* harian yang beragam ("Daily Mix"), dan stasiun "Radio" yang dibuat secara dinamis berdasarkan lagu atau artis yang sedang didengarkan.<sup>7</sup>
  - **Correctness:** Asumsi: Rekomendasi yang diberikan relevan dengan selera musik pengguna, yang didasarkan pada riwayat pendengaran mereka.
  - **Appropriateness:** Sangat sesuai. Dalam pasar yang jenuh, kemampuan untuk membantu pengguna menemukan "lagu favorit mereka berikutnya" adalah nilai jual yang sangat kuat.
- **AI DJ & Smart Shuffle:** Fitur penemuan konten yang lebih interaktif.

- *Completeness*: Fitur Lanjutan. Desain ini mencakup fitur AI DJ yang memberikan komentar kontekstual antar lagu dan *Smart Shuffle* yang secara cerdas menyisipkan lagu-lagu baru yang cocok ke dalam *playlist* yang ada.<sup>7</sup>
- *Correctness*: Asumsi: Transisi AI DJ mulus dan lagu yang disisipkan oleh *Smart Shuffle* sesuai dengan nuansa *playlist*.
- *Appropriateness*: Sesuai. Fitur-fitur ini meningkatkan pengalaman mendengarkan pasif (*lean-back experience*), di mana pengguna ingin menikmati musik tanpa harus aktif memilih setiap lagu.

#### Fitur Sosial dan Komunitas:

Fitur-fitur ini mengubah pengalaman mendengarkan yang soliter menjadi aktivitas komunal.

- **Integrasi Media Sosial**: Kemampuan untuk berbagi aktivitas mendengarkan.
  - *Completeness*: Lengkap. Pengguna dapat membagikan lagu, album, *playlist*, atau profil artis ke platform eksternal. Desain ini secara khusus mencakup fitur berbagi lirik ke Instagram Stories, sebuah format yang sangat populer.<sup>16</sup>
  - *Correctness*: Asumsi: Proses berbagi berjalan lancar, menghasilkan tautan yang benar dan format visual yang menarik di platform tujuan.
  - *Appropriateness*: Sangat sesuai. Fitur ini berfungsi sebagai pemasaran organik (*word-of-mouth*) dan memperkuat identitas sosial pengguna melalui musik.
- **Sesi Grup (Group Session)**: Mendengarkan musik bersama secara *real-time*.
  - *Completeness*: Fitur Inovatif. Desain memungkinkan satu pengguna untuk memulai "Sesi Grup" dan mengundang teman-teman untuk mendengarkan *playlist* yang sama secara sinkron, dengan kontrol pemutaran yang dapat dibagikan, meskipun mereka berada di lokasi yang berbeda.<sup>19</sup>
  - *Correctness*: Asumsi: Sinkronisasi pemutaran antar peserta akurat dan latensinya rendah.
  - *Appropriateness*: Sesuai. Ini menciptakan pengalaman komunal yang unik dan memperkuat aspek sosial dari platform.

Secara keseluruhan, fungsionalitas yang dirancang untuk "Musik Jamak" dapat dipahami dalam sebuah hierarki kebutuhan. Pada tingkat dasar adalah **Fungsionalitas Inti** (pemutaran, pencarian), yang merupakan prasyarat mutlak—seperti fondasi sebuah bangunan. Di atasnya terletak lapisan **Fungsionalitas Personalisasi** (*playlist*, rekomendasi AI), yang memberikan nilai tambah dan membuat pengalaman menjadi milik pengguna secara pribadi. Puncak dari hierarki ini adalah **Fungsionalitas Komunitas** (berbagi, sesi grup), yang mengubah musik menjadi pengalaman bersama. Dalam lanskap kompetitif saat ini, keunggulan sebuah layanan *streaming* tidak lagi ditentukan oleh kekuatan fondasinya—karena fungsionalitas inti telah menjadi komoditas—tetapi oleh kecanggihan dan daya tarik lapisan personalisasi dan komunitas. Oleh karena itu, dalam evaluasi *Functional Appropriateness*, bobot yang lebih besar harus diberikan pada fitur-fitur di lapisan atas ini. Kegagalan dalam memberikan rekomendasi yang relevan, misalnya, akan jauh lebih merusak bagi retensi pengguna jangka

panjang daripada *bug minor* pada fungsi kontrol volume.

### 2.3. Verifikasi Fungsionalitas: Skenario Pengujian Kotak Hitam (Black Box Testing)

Untuk memverifikasi *Functional Correctness* dari desain "Musik Jamak" secara empiris, metodologi *Black Box Testing* diterapkan. Teknik pengujian ini berfokus pada validasi fungsionalitas sistem dari luar, dengan memeriksa apakah output yang dihasilkan sesuai dengan input yang diberikan, tanpa memerlukan pengetahuan tentang struktur kode internal atau arsitektur sistem.<sup>25</sup> Pendekatan ini sangat relevan untuk tahap evaluasi desain karena mensimulasikan interaksi pengguna akhir dan menilai apakah sistem berperilaku sesuai dengan spesifikasi fungsional yang diharapkan.

Tabel berikut menyajikan serangkaian kasus uji (*test cases*) yang dirancang untuk memverifikasi fungsionalitas kritis dari aplikasi "Musik Jamak". Setiap kasus uji mencakup skenario positif (penggunaan normal), dan jika relevan, skenario negatif atau pengujian nilai batas untuk memastikan ketahanan sistem.

**Tabel 1: Kasus Uji Fungsionalitas Kritis "Musik Jamak"**

ID Kasus Uji	Fitur yang Diuji	Skenario Pengujian	Langkah-langkah	Data Uji	Hasil yang Diharapkan
TC-F01	Pencarian	Pengguna mencari artis yang ada (Skenario Positif)	1. Buka aplikasi. 2. Navigasi ke tab 'Cari'. 3. Ketik nama artis di kolom pencarian.	Coldplay	Halaman artis 'Coldplay' ditampilkan, beserta daftar album dan lagu-lagu populer. <sup>27</sup>
TC-F02	Pencarian	Pengguna	1. Buka	Coldpaly	Sistem menampilkan

		mencari dengan kesalahan ketik minor (Skenario Negatif)	aplikasi. 2. Navigasi ke tab 'Cari'. 3. Ketik nama artis dengan kesalahan.		n hasil "Mungkin yang Anda maksud: Coldplay" atau langsung menampilkan hasil yang benar berkat <i>fuzzy matching</i> . <sup>17</sup>
TC-F03	Pemutaran	Pengguna memutar lagu dari hasil pencarian	1. Lakukan langkah-langkah di TC-F01. 2. Ketuk pada lagu "Yellow".	-	Lagu "Yellow" mulai diputar. Antarmuka 'Sedang Diputar' ( <i>Now Playing</i> ) muncul dengan informasi lagu dan kontrol pemutaran. <small>27</small>
TC-F04	Manajemen Playlist	Pengguna menambahkan lagu ke playlist baru	1. Lakukan langkah-langkah di TC-F03. 2. Ketuk ikon '...' pada antarmuka 'Sedang Diputar'. 3.	Nama Playlist: My Favorites	Playlist baru "My Favorites" berhasil dibuat dan lagu "Yellow" ditambahkan ke

			Pilih opsi 'Tambahkan ke Playlist'. 4. Pilih 'Buat playlist baru'. 5. Masukkan nama playlist.		dalamnya. Sebuah notifikasi toast "Ditambahkan ke My Favorites" muncul. <sup>27</sup>
TC-F05	Manajemen Playlist	Pengguna menambahkan lagu yang sudah ada ke playlist	1. Ulangi langkah 1-3 pada TC-F04. 2. Pilih playlist "My Favorites" yang sudah ada.	-	Lagu "Yellow" tetap ada di playlist (tidak terduplicasi). Sistem menampilkan pesan "Lagu ini sudah ada di playlist Anda".
TC-F06	Mode Offline	Pengguna mengunduh playlist (Akun Premium)	1. Navigasi ke playlist "My Favorites". 2. Aktifkan toggle 'Unduh'. 3. Tunggu hingga proses selesai. 4. Aktifkan mode pesawat	-	Indikator progres unduhan muncul dan selesai. Lagu-lagu dalam playlist dapat diputar dengan lancar meskipun perangkat

			pada perangkat. 5. Coba putar lagu dari playlist tersebut.		dalam mode pesawat. <sup>28</sup>
TC-F07	Mode Offline	Pengguna gratis mencoba mengunduh playlist (Skenario Negatif)	1. Login dengan akun gratis. 2. Navigasi ke sebuah playlist. 3. Cari dan coba aktifkan <i>toggle</i> 'Unduh'.	-	Tombol 'Unduh' ditampilkan tetapi tidak aktif ( <i>disabled</i> ) atau saat diketuk, muncul <i>pop-up</i> yang menawarkan untuk <i>upgrade</i> ke akun Premium.

Tabel ini menunjukkan bagaimana pendekatan pengujian yang terstruktur dapat mengubah klaim kualitatif tentang fungsionalitas menjadi hasil yang dapat diverifikasi. Dengan mencakup berbagai skenario, evaluasi ini memberikan gambaran yang lebih lengkap dan kuat tentang kebenaran dan ketahanan fungsional dari desain yang diusulkan.

## Bagian 3: Analisis Evaluasi Usability

### 3.1. Kerangka Evaluasi: 10 Heuristik Usability Nielsen

Evaluasi *usability* bertujuan untuk mengukur seberapa mudah, efisien, dan memuaskan sebuah antarmuka digunakan oleh pengguna akhir. Untuk melakukan evaluasi ini secara sistematis, laporan ini menggunakan **10 Heuristik Usability** yang dikembangkan oleh Jakob Nielsen. Heuristik ini bukanlah pedoman desain yang kaku, melainkan serangkaian prinsip umum atau "aturan praktis" yang berasal dari riset bertahun-tahun terhadap perilaku pengguna. Kerangka kerja ini telah menjadi standar industri untuk melakukan *heuristic evaluation*, sebuah metode inspeksi *usability* di mana evaluator ahli memeriksa antarmuka dan menilai kepatuhannya terhadap prinsip-prinsip ini untuk mengidentifikasi potensi masalah *usability*.<sup>3</sup>

Ke-10 heuristik tersebut adalah:

1. **Visibility of system status (Visibilitas status sistem)**: Sistem harus selalu memberi tahu pengguna tentang apa yang sedang terjadi melalui umpan balik yang sesuai dalam waktu yang wajar.
2. **Match between system and the real world (Kecocokan antara sistem dan dunia nyata)**: Sistem harus berbicara dalam bahasa pengguna, menggunakan kata-kata, frasa, dan konsep yang akrab bagi pengguna, bukan istilah teknis.
3. **User control and freedom (Kontrol dan kebebasan pengguna)**: Pengguna sering kali melakukan kesalahan. Mereka membutuhkan "pintu darurat" yang jelas untuk keluar dari keadaan yang tidak diinginkan tanpa harus melalui proses yang panjang. Dukungan untuk *undo* dan *redo* sangat penting.
4. **Consistency and standards (Konsistensi dan standar)**: Pengguna tidak seharusnya bertanya-tanya apakah kata, situasi, atau tindakan yang berbeda memiliki arti yang sama. Ikuti konvensi platform.
5. **Error prevention (Pencegahan kesalahan)**: Desain yang baik adalah yang mencegah masalah terjadi sejak awal.
6. **Recognition rather than recall (Mengenali daripada mengingat)**: Minimalkan beban memori pengguna dengan membuat objek, tindakan, dan opsi terlihat. Pengguna tidak harus mengingat informasi dari satu bagian dialog ke bagian lain.
7. **Flexibility and efficiency of use (Fleksibilitas dan efisiensi penggunaan)**: Accelerators—yang tidak terlihat oleh pengguna pemula—dapat mempercepat interaksi bagi pengguna ahli. Sistem harus dapat melayani pengguna berpengalaman dan tidak berpengalaman.
8. **Aesthetic and minimalist design (Desain estetis dan minimalis)**: Dialog tidak boleh berisi informasi yang tidak relevan atau jarang dibutuhkan. Setiap unit informasi tambahan bersaing dengan unit informasi yang relevan dan mengurangi visibilitas relatifnya.
9. **Help users recognize, diagnose, and recover from errors (Bantu pengguna mengenali, mendiagnosis, dan pulih dari kesalahan)**: Pesan kesalahan harus dinyatakan dalam bahasa sederhana, menunjukkan masalah secara tepat, dan secara

konstruktif menyarankan solusi.

10. **Help and documentation (Bantuan dan dokumentasi):** Meskipun lebih baik jika sistem dapat digunakan tanpa dokumentasi, mungkin perlu menyediakan bantuan dan dokumentasi.

Dengan menggunakan sepuluh lensa ini, evaluasi dapat secara komprehensif mengidentifikasi area-area di mana desain antarmuka "Musik Jamak" mungkin menyebabkan kebingungan, frustrasi, atau inefisiensi bagi pengguna.

### 3.2. Evaluasi Heuristik Antarmuka "Musik Jamak"

Analisis berikut menerapkan 10 Heuristik Nielsen pada desain antarmuka "Musik Jamak" di ketiga platform. Temuan disajikan dalam format *checklist* untuk memberikan gambaran yang jelas tentang kepatuhan dan potensi pelanggaran. Setiap pelanggaran yang teridentifikasi diberi peringkat keparahan untuk membantu memprioritaskan rekomendasi perbaikan. Peringkat keparahan menggunakan skala 1 hingga 4: 1 (Kosmetik), 2 (Minor), 3 (Mayor), 4 (Katastropik).

**Tabel 2: Checklist Evaluasi Heuristik "Musik Jamak"**

Heuristik	Pertanyaan Evaluasi Kunci	Analisis Kepatuhan pada "Musik Jamak" (Contoh Positif/Negatif)	Tingkat Keparahan Pelanggaran
<b>1. Visibility of System Status</b>	Apakah pengguna selalu diberi tahu tentang apa yang sedang terjadi pada sistem?	<b>Positif:</b> Saat lagu diputar, <i>progress bar</i> secara jelas menunjukkan posisi waktu saat ini. Saat lagu ditambahkan ke <i>playlist</i> , notifikasi <i>toast</i> muncul sebagai konfirmasi. <sup>29</sup> Saat	2 (Minor)

		<p>mengunduh, ikon progres melingkar menunjukkan status unduhan.</p> <p><b>Negatif:</b> Sinkronisasi <i>playlist</i> antar perangkat terjadi di latar belakang tanpa indikator yang jelas. Pengguna mungkin bingung mengapa <i>playlist</i> yang baru dibuat di desktop belum muncul di ponsel.</p>	
<b>2. Match between System and Real World</b>	Apakah sistem menggunakan bahasa dan konsep yang akrab bagi pengguna?	<p><b>Positif:</b> Menggunakan ikonografi yang dikenal secara universal: ikon 'hati' untuk 'suka', 'keranjang belanja' (jika ada <i>merchandise</i>), dan 'tempat sampah' untuk menghapus. Konsep '<i>playlist</i>' dan '<i>album</i>' meniru koleksi musik fisik.[30, 31]</p>	N/A
<b>3. User Control and Freedom</b>	Apakah pengguna dapat dengan mudah membatalkan aksi atau keluar dari keadaan yang tidak	<p><b>Positif:</b> Pengguna dapat dengan mudah menghapus lagu dari <i>playlist</i> atau berhenti mengikuti seorang</p>	3 (Mayor)

	diinginkan?	<p>artis. Tombol 'kembali' selalu tersedia untuk navigasi mundur.</p> <p><b>Negatif:</b> Tidak ada fungsi "Undo" setelah secara tidak sengaja menghapus sebuah <i>playlist</i> utuh. Ini adalah aksi destruktif tanpa jalan kembali yang mudah.<sup>29</sup></p>	
<b>4. Consistency and Standards</b>	Apakah elemen dan alur kerja konsisten di seluruh aplikasi dan mengikuti konvensi platform?	<p><b>Positif:</b> Tata letak navigasi utama (Beranda, Cari, Koleksi) konsisten di bagian bawah layar pada aplikasi <i>mobile</i>, sesuai dengan pedoman desain iOS dan Android.<sup>32</sup> Ikon dan terminologi yang sama digunakan secara konsisten di semua platform.</p>	N/A
<b>5. Error Prevention</b>	Apakah desain membantu mencegah pengguna melakukan kesalahan?	<p><b>Positif:</b> Tombol 'Buat Playlist' dinonaktifkan (<i>disabled</i>) hingga pengguna memasukkan setidaknya satu karakter untuk nama <i>playlist</i>,</p>	2 (Minor)

		<p>mencegah pembuatan <i>playlist</i> tanpa nama.<sup>32</sup></p> <p><b>Negatif:</b> Dalam <i>playlist</i> kolaboratif, tidak ada dialog konfirmasi saat seorang kolaborator menghapus lagu yang ditambahkan oleh orang lain. Ini dapat menyebabkan konflik sosial yang tidak disengaja.</p>	
<b>6. Recognition Rather Than Recall</b>	Apakah antarmuka meminimalkan beban memori pengguna dengan membuat opsi dan informasi terlihat?	<p><b>Positif:</b> Menampilkan sampul album di samping judul lagu sangat membantu pengenalan visual.<sup>3</sup> Daftar "Recently Played" dan "Search History" berfungsi sebagai pengingat, sehingga pengguna tidak perlu mengingat apa yang mereka dengarkan atau cari sebelumnya.<sup>18</sup></p>	N/A
<b>7. Flexibility and Efficiency of Use</b>	Apakah sistem efisien untuk pengguna pemula	<p><b>Positif:</b> Pengguna pemula dapat dengan mudah menavigasi melalui</p>	N/A

	dan ahli?	antarmuka grafis. Untuk pengguna ahli di platform desktop, tersedia <i>keyboard shortcuts</i> (hotkeys) untuk mengontrol pemutaran (play/pause, next, volume) tanpa harus menggunakan mouse, yang meningkatkan efisiensi.[33, 34]	
<b>8. Aesthetic and Minimalist Design</b>	Apakah antarmuka bebas dari informasi atau elemen visual yang tidak relevan?	<b>Positif:</b> Layar 'Sedang Diputar' ( <i>Now Playing</i> ) sangat minimalis, memfokuskan perhatian pada sampul album, judul lagu, dan kontrol esensial. Opsi sekunder seperti 'Bagikan', 'Tambahkan ke Antrean', atau 'Sleep Timer' disembunyikan di dalam menu '...' untuk menjaga kebersihan visual.[19, 35]	N/A
<b>9. Help Users Recover from Errors</b>	Apakah pesan kesalahan jelas, informatif, dan menawarkan	<b>Negatif:</b> Ketika pemutaran gagal karena tidak ada koneksi internet,	3 (Mayor)

	solusi?	<p>aplikasi hanya menampilkan pesan kesalahan generik seperti "Gagal memuat konten". Pesan ini tidak secara eksplisit menyarankan solusi, seperti "Periksa koneksi internet Anda" atau "Coba putar musik yang telah diunduh".[35, 36]</p>	
<b>10. Help and Documentation</b>	Apakah bantuan dan dokumentasi mudah ditemukan dan relevan dengan tugas pengguna?	<p><b>Positif:</b> Terdapat bagian "Bantuan" atau "FAQ" yang dapat diakses dari menu pengaturan.</p> <p><b>Negatif:</b> Bantuan bersifat pasif. Tidak ada bantuan kontekstual (<i>contextual help</i>), seperti ikon tanda tanya kecil di sebelah fitur yang kompleks (misalnya, pengaturan <i>crossfade</i>) yang dapat menjelaskan fungsinya secara langsung tanpa harus meninggalkan layar saat ini.</p>	2 (Minor)

Analisis heuristik ini mengubah kritik *usability* yang seringkali subjektif menjadi serangkaian temuan yang terstruktur dan dapat ditindaklanjuti. Dengan mengidentifikasi pelanggaran spesifik dan menilai tingkat keparahannya, laporan ini dapat memberikan dasar yang kuat untuk rekomendasi perbaikan desain yang terprioritaskan, memastikan bahwa upaya pengembangan difokuskan pada masalah yang paling berdampak pada pengalaman pengguna.

## Bagian 4: Analisis Komparatif dan Rekomendasi Strategis

### 4.1. Perbandingan Fungsionalitas dan Usability Lintas Platform

Sebuah aplikasi modern yang sukses harus memberikan pengalaman yang konsisten namun tetap dioptimalkan untuk konteks penggunaan setiap platform. "Musik Jamak", yang meniru Spotify, dirancang untuk tiga lingkungan yang berbeda: Desktop, Mobile, dan Web. Masing-masing memiliki karakteristik, batasan, dan ekspektasi pengguna yang unik. Analisis komparatif ini mensintesis temuan dari evaluasi fungsionalitas dan *usability* untuk menyoroti perbedaan strategis dalam desain lintas platform.

Tabel berikut menyajikan perbandingan fitur dan aspek pengalaman pengguna secara berdampingan. Tujuannya adalah untuk memvisualisasikan bagaimana desain "Musik Jamak" beradaptasi dengan setiap platform, menonjolkan *trade-off* yang dibuat, dan mengevaluasi koherensi strategi produk secara keseluruhan.

**Tabel 3: Perbandingan Fitur dan Pengalaman Pengguna Lintas Platform "Musik Jamak"**

Fitur/Aspek	Versi Desktop	Versi Mobile	Versi Web	Analisis Strategis
Kualitas Audio	Kualitas tertinggi (hingga 320 kbps), opsi	Kualitas tinggi (hingga 320 kbps), opsi	Kualitas terbatas (hingga 256 kbps)	Desktop diposisikan

<b>(Premium)</b>	kbps Ogg Vorbis/AAC), kontrol equalizer penuh. <sup>33</sup>	kualitas terpisah untuk Wi-Fi dan data seluler, equalizer terbatas.[37, 38]	kbps AAC), tanpa equalizer.[39, 40]	sebagai platform untuk pengalaman mendengarkan dengan fidelitas tertinggi ( <i>audiophile-centric</i> ). <b>Mobile</b> menyeimbangkan kualitas dengan efisiensi data, sebuah pertimbangan krusial untuk penggunaan saat bepergian. <b>Web</b> menawarkan kemudahan akses dengan kualitas audio yang "cukup baik" untuk pendengar kasual.
<b>Mode Offline</b>	Ya, dapat mengunduh <i>playlist</i> dan album untuk pemutaran tanpa koneksi. <sup>33</sup>	Ya, fitur ini merupakan salah satu nilai jual utama untuk penggunaan <i>mobile</i> .[38, 39]	Tidak tersedia. Memerlukan koneksi internet aktif. <sup>37</sup>	Perbedaan ini mencerminkan konteks penggunaan yang paling umum. Mode <i>offline</i> sangat penting di <b>Mobile</b> karena

				konektivitas yang tidak stabil dan untuk menghemat data. Di <b>Desktop</b> , fitur ini berguna tetapi kurang krusial. Ketiadaannya di <b>Web</b> dapat diterima karena asumsi konektivitas yang konstan.
<b>Manajemen Perpustakaan</b>	Fitur paling lengkap. Termasuk kemampuan untuk membuat <b>folder</b> untuk mengorganisir <i>playlist</i> dan mengimpor file musik lokal dari komputer.[33, 34]	Manajemen <i>playlist</i> dan album penuh, tetapi tanpa fitur folder atau impor file lokal. <sup>33</sup>	Manajemen <i>playlist</i> dasar, tanpa fitur folder atau impor file lokal. <sup>33</sup>	<b>Desktop</b> berfungsi sebagai pusat manajemen perpustakaan ( <i>library hub</i> ) untuk power users dengan koleksi yang besar dan terorganisir. <b>Mobile</b> dan <b>Web</b> menyediakan akses dan manajemen yang lebih sederhana, difokuskan pada konsumsi daripada organisasi

				mendalam.
<b>Interaksi Utama &amp; Efisiensi</b>	Dioptimalkan untuk mouse dan keyboard. Mendukung <b>hotkeys</b> (pintasan keyboard) untuk kontrol pemutaran yang cepat, sangat efisien untuk multitasking.[33, 34]	Dioptimalkan untuk sentuhan (touch gestures). Navigasi utama di bagian bawah layar untuk jangkauan ibu jari yang mudah.	Dioptimalkan untuk klik mouse. Antarmuka yang sederhana dan langsung, tanpa memerlukan pembelajaran pintasan.	Setiap platform dioptimalkan untuk metode input utamanya. Ketiadaan <b>hotkeys</b> di <b>Web</b> adalah sebuah <i>trade-off</i> untuk kesederhanaan dan aksesibilitas universal, sementara kehadirannya di <b>Desktop</b> melayani pengguna yang menggunakan aplikasi dalam alur kerja mereka.
<b>Konsumsi Sumber Daya</b>	Konsumsi RAM dan CPU bisa signifikan, terutama saat aplikasi berjalan lama. Menawarkan akselerasi perangkat keras ( <i>hardware</i>	Dioptimalkan untuk efisiensi baterai dan data. Memiliki mode 'Penghemat Data' ( <i>Data Saver</i> ). <sup>37</sup>	Tergantung pada efisiensi <i>browser</i> . Umumnya lebih ringan dari aplikasi desktop tetapi bisa mengonsumsi sumber daya <i>browser</i>	Prioritas desain sangat jelas. <b>Mobile</b> mengutamakan penghematan sumber daya. <b>Desktop</b> mengutamakan performa dan kekayaan

	acceleration).[33, 40]		secara signifikan.	fitur, dengan asumsi sumber daya yang lebih melimpah. <b>Web</b> berada di tengah, bergantung pada platform eksternal ( <i>browser</i> ).
<b>Aksesibilitas &amp; Instalasi</b>	Memerlukan instalasi perangkat lunak.	Memerlukan instalasi dari App Store/Play Store.	Tidak memerlukan instalasi. Dapat diakses dari <i>browser</i> apa pun, membuatnya sangat portabel dan ideal untuk komputer publik atau bersama.[33, 39]	<b>Web Player</b> unggul dalam hal kemudahan akses awal dan portabilitas. Ini adalah titik masuk tanpa friksi bagi pengguna baru. <b>Desktop</b> dan <b>Mobile</b> menawarkan pengalaman yang lebih terintegrasi dan kaya fitur dengan imbalan proses instalasi.

Tabel perbandingan ini secara jelas mengilustrasikan bahwa desain "Musik Jamak" tidak menerapkan pendekatan "satu ukuran untuk semua". Sebaliknya, ini adalah contoh desain yang sadar konteks (*context-aware design*). Platform **Desktop** dirancang sebagai alat utama bagi pengguna yang paling berdedikasi, menawarkan fitur manajemen dan kualitas audio terbaik. Platform **Mobile** adalah pendamping setia saat bepergian, dengan prioritas utama pada efisiensi dan aksesibilitas *offline*. Sementara itu, **Web Player** berfungsi sebagai gerbang

akses universal yang cepat dan mudah, mengorbankan beberapa fitur canggih demi kenyamanan maksimal. Koherensi strategi ini memastikan bahwa pengguna mendapatkan pengalaman yang paling sesuai dengan perangkat dan situasi mereka saat itu.

## 4.2. Rekomendasi Perbaikan Desain

Berdasarkan analisis komprehensif terhadap fungsionalitas dan *usability*, berikut adalah serangkaian rekomendasi yang konkret, dapat ditindaklanjuti, dan diprioritaskan untuk meningkatkan kualitas desain perangkat lunak "Musik Jamak". Rekomendasi ini dikelompokkan berdasarkan area evaluasi dan diurutkan berdasarkan tingkat prioritas, yang ditentukan oleh dampak potensialnya terhadap pengalaman pengguna dan keberhasilan produk.

### Rekomendasi Fungsionalitas (Berdasarkan Tabel 1):

- **Prioritas Tinggi - Perbaikan Fungsionalitas Inti:**
  - **Rekomendasi F-1:** Implementasikan mekanisme penanganan duplikasi lagu dalam *playlist* yang lebih baik. Berdasarkan potensi kegagalan pada **TC-F05**, alih-alih hanya menampilkan pesan kesalahan, sistem seharusnya memberikan opsi kepada pengguna, seperti "Lagu ini sudah ada di playlist. Tambahkan lagi?". Ini memberikan kontrol lebih kepada pengguna yang mungkin sengaja ingin menduplikasi lagu.
  - **Rekomendasi F-2:** Tingkatkan algoritma pencarian *fuzzy matching*. Berdasarkan **TC-F02**, meskipun sistem dapat menangani kesalahan ketik sederhana, sistem harus diuji dan ditingkatkan untuk menangani variasi yang lebih kompleks (misalnya, transliterasi, nama artis dengan karakter non-Latin) untuk memastikan kebenaran fungsional yang lebih tinggi.

### Rekomendasi Usability (Berdasarkan Tabel 2):

- **Prioritas Tinggi (Mengatasi Pelanggaran Heuristik Mayor - Keparahan 3):**
  - **Rekomendasi U-1 (Terkait Heuristik #3 - User Control and Freedom):** Implementasikan fitur "Undo" untuk aksi penghapusan *playlist*. Menghapus *playlist* adalah aksi destruktif dengan dampak tinggi. Sediakan opsi "Undo" yang muncul selama beberapa detik setelah penghapusan, atau pindahkan *playlist* yang dihapus ke folder "Baru Dihapus" yang dapat dipulihkan dalam 30 hari, mirip dengan sistem file atau aplikasi email modern.
  - **Rekomendasi U-2 (Terkait Heuristik #9 - Help Users Recover from Errors):** Rancang ulang semua pesan kesalahan agar lebih informatif dan konstruktif. Ganti pesan generik seperti "Gagal memuat konten" dengan pesan spesifik yang

mendiagnos masalah dan menyarankan solusi. Contoh: "Koneksi internet terputus. Silakan periksa pengaturan jaringan Anda atau beralih ke musik yang telah diunduh untuk didengarkan secara offline."

- **Prioritas Sedang (Mengatasi Pelanggaran Heuristik Minor - Keparahan 2):**
  - **Rekomendasi U-3 (Terkait Heuristik #1 - Visibility of System Status):** Tambahkan indikator status sinkronisasi yang jelas namun tidak mengganggu. Sebuah ikon kecil di *header* aplikasi yang beranimasi saat sinkronisasi aktif dan menampilkan tanda centang saat selesai akan memberikan umpan balik yang dibutuhkan pengguna tanpa mengacaukan antarmuka.
  - **Rekomendasi U-4 (Terkait Heuristik #5 - Error Prevention):** Tambahkan dialog konfirmasi untuk tindakan penting dalam konteks kolaboratif. Saat seorang pengguna mencoba menghapus lagu dari *playlist* kolaboratif, tampilkan dialog konfirmasi seperti: "Anda akan menghapus '[Nama Lagu]' dari *playlist* ini. Tindakan ini akan memengaruhi semua kolaborator. Lanjutkan?".
  - **Rekomendasi U-5 (Terkait Heuristik #10 - Help and Documentation):** Implementasikan bantuan kontekstual. Untuk fitur-fitur yang mungkin tidak langsung dipahami oleh semua pengguna (misalnya, *Crossfade*, *Audio Normalization*), tambahkan ikon 'info' (i) kecil di sebelahnya. Saat diketuk, ikon ini akan menampilkan *tooltip* atau *pop-up* singkat yang menjelaskan fungsi fitur tersebut secara langsung.

#### **Rekomendasi Strategis Lintas Platform (Berdasarkan Tabel 3):**

- **Rekomendasi P-1:** Pertimbangkan untuk membawa fitur organisasi *playlist* tingkat lanjut (folder) ke platform *mobile*. Mengingat pengguna *mobile* juga bisa menjadi *power users*, fitur ini dapat diimplementasikan sebagai opsi lanjutan di dalam halaman 'Koleksi Anda', sehingga tidak mengganggu pengguna kasual tetapi memberdayakan mereka yang memiliki banyak *playlist*.
- **Rekomendasi P-2:** Tingkatkan kualitas audio pada Web Player. Meskipun aksesibilitas adalah kekuatan utamanya, kesenjangan kualitas audio dengan aplikasi *native* terlalu besar. Jelajahi kemungkinan untuk mendukung *codec* yang lebih efisien atau *bitrate* yang lebih tinggi untuk pengguna Premium di web, setidaknya mendekati kualitas aplikasi *native*, untuk memberikan pengalaman yang lebih konsisten bagi pelanggan berbayar.

### **4.3. Kesimpulan**

Laporan evaluasi ini telah melakukan analisis mendalam terhadap desain perancangan perangkat lunak "Musik Jamak" dari dua pilar kualitas utama: fungsionalitas dan *usability*. Melalui pemodelan sistem dengan UML, penerapan kerangka kerja formal ISO/IEC 25010, dan evaluasi heuristik Nielsen, sebuah gambaran komprehensif mengenai kekuatan dan

kelemahan desain telah berhasil dipetakan.

Temuan utama menunjukkan bahwa desain "Musik Jamak" memiliki fondasi yang kuat. Secara **fungsional**, ia mencakup serangkaian fitur inti, personalisasi, dan sosial yang komprehensif, sejajar dengan standar industri yang ditetapkan oleh Spotify. Desain ini menunjukkan pemahaman yang baik tentang hierarki kebutuhan pengguna dalam layanan *streaming* musik modern, di mana personalisasi dan komunitas menjadi faktor pembeda utama. Dari segi **usability**, desain antarmuka secara umum mematuhi prinsip-prinsip desain yang solid, menghasilkan pengalaman yang estetis, minimalis, dan konsisten di berbagai platform. Kekuatan utamanya terletak pada pengenalan visual yang kuat, efisiensi penggunaan bagi berbagai tingkat keahlian pengguna, dan tata letak yang bersih.

Meskipun demikian, evaluasi ini juga mengidentifikasi beberapa area kritis yang memerlukan perbaikan. Kelemahan utama dalam desain terletak pada penanganan kesalahan dan pemulihan. Kurangnya opsi "Undo" untuk tindakan destruktif dan pesan kesalahan yang tidak informatif merupakan pelanggaran heuristik mayor yang dapat menyebabkan frustrasi signifikan bagi pengguna. Selain itu, terdapat inkonsistensi minor dalam visibilitas status sistem, terutama terkait proses latar belakang seperti sinkronisasi, yang dapat menimbulkan kebingungan.

Analisis komparatif lintas platform menyoroti keberhasilan strategi desain yang sadar konteks, di mana setiap versi (Desktop, Mobile, Web) dioptimalkan untuk lingkungan penggunaannya masing-masing. Namun, ini juga mengungkapkan adanya kesenjangan fitur yang mungkin perlu dievaluasi kembali, seperti ketiadaan alat organisasi *playlist* canggih di *mobile* dan kualitas audio yang lebih rendah di web.

Pada akhirnya, laporan ini menyimpulkan bahwa desain "Musik Jamak" merupakan rancangan yang matang dan berpotensi besar. Kekuatan fundamentalnya jauh melampaui kelemahannya. Dengan mengimplementasikan rekomendasi yang telah diuraikan—terutama dalam meningkatkan mekanisme pemulihan dari kesalahan dan memperkaya fitur pada platform *mobile*—"Musik Jamak" dapat bertransformasi dari sekadar replika fungsional menjadi sebuah produk dengan pengalaman pengguna yang unggul dan koheren. Keberhasilan akhir dari aplikasi semacam ini tidak hanya terletak pada peniruan fitur, tetapi pada eksekusi desain yang cermat dan pemahaman mendalam tentang nuansa interaksi pengguna dalam setiap konteks yang unik.