

**ULANGAN AKHIR SEMESTER**

**“OPEN YOUR PODCAST”**



**Pemograman Web**

**CR002**

Nama:

Firnanda Amalia (20230803040)

Dosen Pengampu:

Jefry Sunupurwa Asri, S.Kom.,M.Kom

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS ESA UNGGUL**

**2025**

## **Aplikasi Voting Ide Podcast**

### **1. Pendahuluan**

#### **1.1. Tujuan**

Dokumen ini menjelaskan spesifikasi perangkat lunak aplikasi Voting Ide Podcast yang memungkinkan pengguna untuk memilih ide podcast yang disediakan admin. Dokumen ini menjadi acuan dalam pengembangan, implementasi, dan evaluasi aplikasi.

### **2. Lingkup Sistem**

Aplikasi berbasis web yang memungkinkan:

- Pengguna umum memberikan suara pada ide podcast.
- Admin mengelola ide & melihat hasil voting melalui dashboard admin.
- Sistem sederhana dengan fitur dasar yang dapat berjalan di web browser.

### **3. Definisi, Akronim, dan Singkatan**

- **Admin:** Pengelola aplikasi.
- **Pengguna:** Orang yang memilih ide podcast.
- **Filament:** Laravel admin dashboard untuk admin.
- **Blade:** Template engine Laravel untuk frontend.
- **CRUD:** Create, Read, Update, Delete.

### **4. Deskripsi Umum**

#### **4.1. Pengguna**

- **Pengguna Umum:** Bisa memilih satu ide dengan memasukkan namanya & memilih ide. Tidak perlu akun.
- **Admin:** Dapat melihat ide, jumlah suara, daftar voter, dan melakukan CRUD pada ide & voting.

#### **4.2. Batasan**

- Hanya 1 suara per pengguna per ide.
- Login hanya untuk admin.
- Tidak ada fitur registrasi/pengaturan pengguna umum.

- Fokus pada fungsionalitas dasar.

## 5. Fitur Fungsional

Fitur	Deskripsi	Input	Output
Voting (User)	Memilih ide dengan mengisi nama & klik vote	Nama voter + ide terpilih	Status “vote berhasil”
Dashboard Admin – Ideas	Melihat daftar ide & suara, CRUD ide	Judul ide, deskripsi	Daftar ide dengan jumlah suara
Dashboard Admin – Votes	Melihat siapa memilih apa, CRUD vote	Nama voter, ide terpilih	Daftar votes
API	Endpoint untuk daftar ide & top votes	GET/POST	JSON data

## 6. Antarmuka/interface

### 6.1. Frontend

Gambar 3: Halaman Voting (User)

Fitur: Daftar ide + suara + form voting (nama + pilih ide + tombol vote).

The screenshot displays a web interface titled "Daftar Podcast". It contains six voting cards arranged in a 2x3 grid. Each card has a title, a short description, the current number of votes, a text input field for the user's name, and a blue "Vote" button.

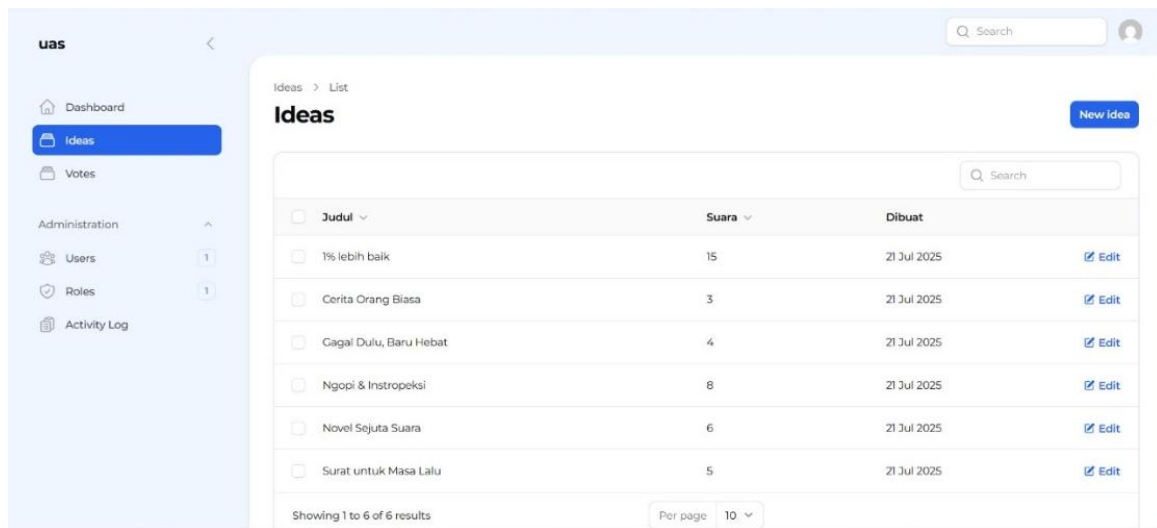
- Card 1:** Title "1% lebih baik", Description "Menjadi lebih baik dari sebelumnya", Votes: 15.
- Card 2:** Title "Cerita Orang Biasa", Description "Mewawancarai orang biasa dengan pengalaman luar biasa", Votes: 3.
- Card 3:** Title "Gagal Dulu, Baru Hebat", Description "Kisah inspiratif dari kegagalan menuju kesuksesan", Votes: 4.
- Card 4:** Title "Ngopi & Instropeksi", Description "Curhat monolog soal keresahan anak muda (quarter-life crisis, overthinking, dll)", Votes: 8.
- Card 5:** Title "Novel Sejuta Suara", Description "Tiap episode lanjut cerita fiksi bersambung, dibacakan oleh host dan guest", Votes: 6.
- Card 6:** Title "Surat untuk Masa Lalu", Description "Baca dan bahas surat terbuka yang ditujukan ke diri sendiri di masa lalu", Votes: 5.

### 6.2. Backend

Gambar 1: Dashboard Admin → Ideas

memberi admin kontrol penuh atas pengelolaan ide yang ditampilkan dalam sistem, termasuk pemantauan jumlah suara yang diterima oleh setiap ide.

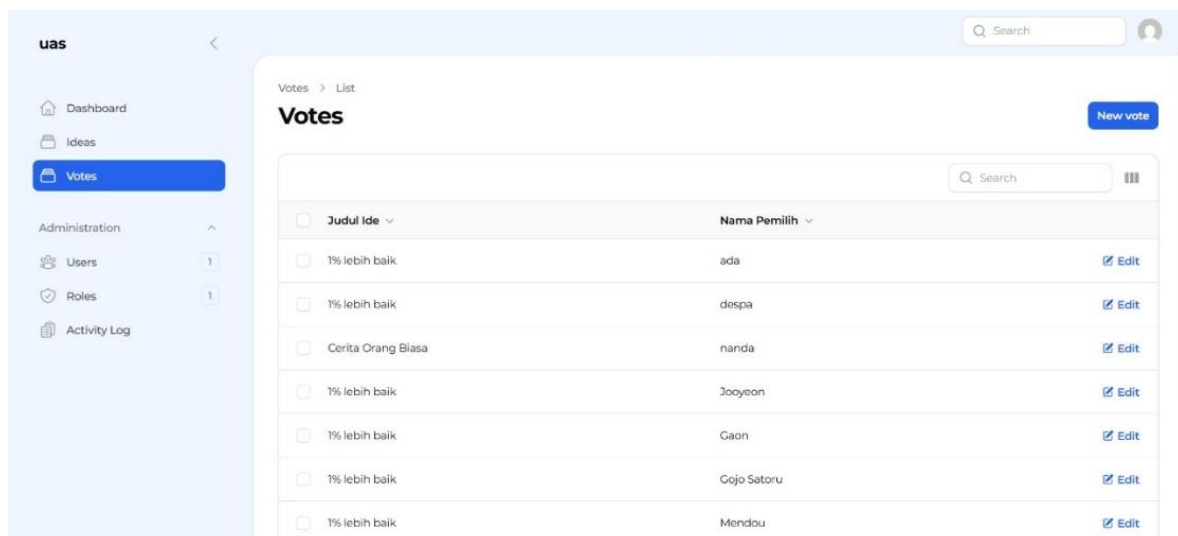
Fitur: CRUD ide & melihat jumlah suara.



**Gambar 2: Dashboard Admin → Votes**

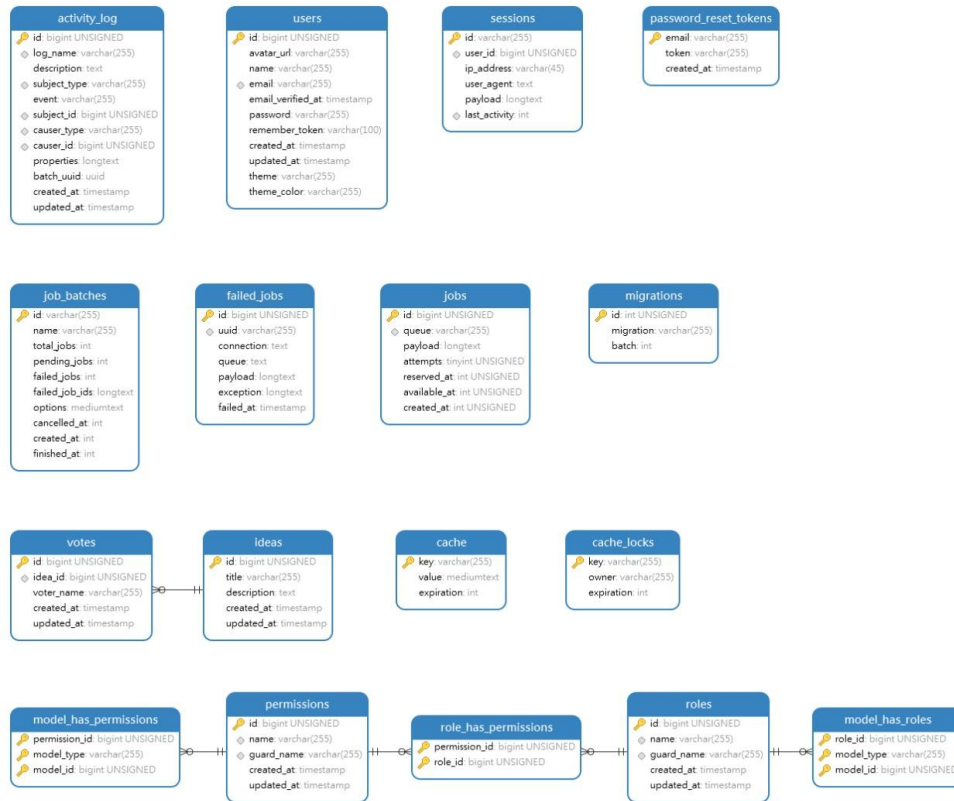
Votes memungkinkan admin untuk melihat siapa saja yang telah memberikan suara, serta melakukan modifikasi terhadap data suara jika diperlukan.

Fitur: CRUD vote, lihat siapa memilih apa.



## 7. Database

Diagram *Entity Relationship Diagram (ERD)*



### Entitas dan Relasi dalam Diagram ERD

#### 1. Users

- **Fields:** id, avatar\_url, name, email, email\_verified\_at, password, remember\_token, created\_at, updated\_at
- Menyimpan informasi pengguna sistem, termasuk nama, email, password, dan avatar.
- **Relasi:**
  - **Sessions:** Setiap pengguna dapat memiliki sesi login di tabel sessions.
  - **Votes:** Setiap pengguna dapat memberikan suara pada ide, yang akan disimpan di tabel votes.

#### 2. Sessions

- **Fields:** id, user\_id, ip\_address, user\_agent, payload, last\_activity
- Menyimpan informasi sesi pengguna yang sedang aktif, termasuk ID pengguna, alamat IP, dan user agent.

- **Relasi:** Terhubung ke **Users** melalui user\_id.

### 3. Votes

- **Fields:** id, idea\_id, user\_id, voter\_name, created\_at, updated\_at
- Menyimpan data suara yang diberikan oleh pengguna untuk setiap ide podcast.
- **Relasi:**
  - Terhubung ke **Users** melalui user\_id untuk melacak siapa yang memberikan suara.
  - Terhubung ke **Ideas** melalui idea\_id untuk menunjukkan ide mana yang dipilih.

### 4. Ideas

- **Fields:** id, title, description, created\_at, updated\_at
- Menyimpan informasi tentang ide podcast, seperti judul dan deskripsi.
- **Relasi:**
  - Terhubung ke **Votes** melalui id yang menjadi idea\_id di tabel **Votes**.

### 5. Cache

- **Fields:** key, value, expiration
- Menyimpan data sementara untuk mengoptimalkan proses, dengan waktu kadaluarsa.
- **Relasi:** Tidak ada relasi langsung dengan entitas lain dalam diagram ini.

### 6. Cache\_Locks

- **Fields:** key, owner, expiration
- Digunakan untuk mengunci cache untuk memastikan tidak ada proses yang bersaing dalam mengakses data sementara.
- **Relasi:** Tidak ada relasi langsung dengan entitas lain dalam diagram ini.

### 7. Activity Log

- **Fields:** id, log\_name, description, subject\_type, event, subject\_id, causer\_type, causer\_id, batch\_uuid, created\_at, updated\_at
- Menyimpan log aktivitas pengguna dan admin di sistem.

- **Relasi:** Tidak ada relasi langsung dengan entitas lain dalam diagram ini.

## 8. Permissions

- **Fields:** id, name, guard\_name, created\_at, updated\_at
- Menyimpan data terkait dengan izin akses dalam sistem.
- **Relasi:**
  - **Role\_has\_permissions:** Entitas ini terhubung ke **Roles** untuk menentukan peran yang memiliki izin tertentu.
  - **Model\_has\_permissions:** Digunakan untuk mengaitkan izin dengan model yang sesuai, seperti pengguna.

## 9. Roles

- **Fields:** id, name, guard\_name, created\_at, updated\_at
- Menyimpan data tentang peran (roles) dalam sistem (misalnya admin, pengguna biasa).
- **Relasi:**
  - Terhubung dengan **Model\_has\_roles** untuk menentukan peran yang diberikan kepada model tertentu (seperti pengguna).

## 10. Role\_has\_permissions

- **Fields:** permission\_id, role\_id
- Menyimpan hubungan antara peran dan izin yang terkait.
- **Relasi:** Terhubung ke **Roles** dan **Permissions**.

## 11. Model\_has\_roles

- **Fields:** role\_id, model\_type, model\_id
- Menyimpan hubungan antara model dan peran yang diberikan kepadanya.
- **Relasi:** Terhubung ke **Roles** untuk menentukan peran yang diberikan kepada model (misalnya pengguna).

## 12. Password\_Reset\_Tokens

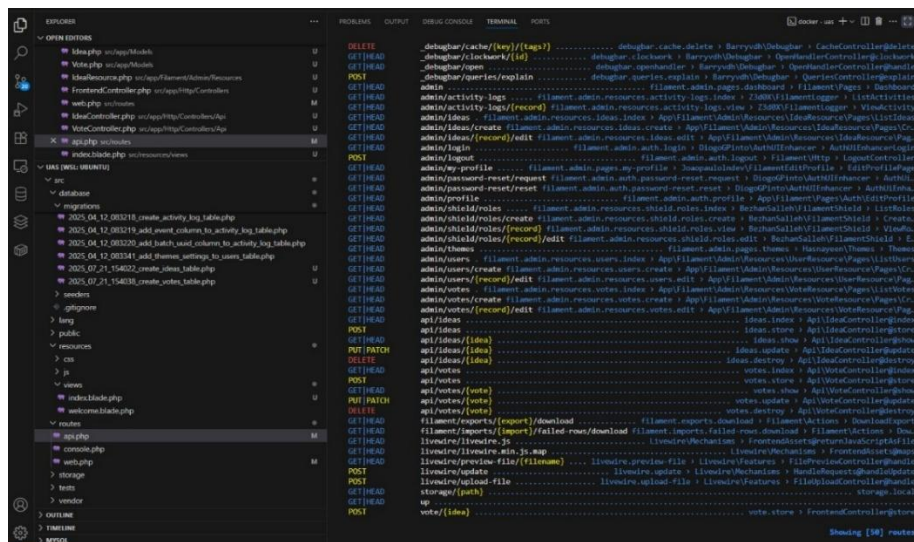
- **Fields:** email, token, created\_at
- Menyimpan token untuk proses reset password.
- **Relasi:** Terhubung dengan **Users** melalui email.

**Relasi Antar Entitas:**

- **Users ↔ Votes:** Relasi ini menggambarkan bahwa setiap pengguna dapat memberikan satu suara pada ide tertentu. Setiap suara mencatat pengguna yang memberi suara dan ide yang dipilih.
- **Ideas ↔ Votes:** Setiap ide dapat menerima banyak suara. Relasi ini menggambarkan hubungan antara ide dan suara yang diberikan oleh pengguna.
- **Roles ↔ Model\_has\_roles:** Relasi ini menghubungkan peran dengan pengguna atau model lainnya.
- **Permissions ↔ Role\_has\_permissions:** Menentukan izin yang dimiliki oleh peran tertentu.

## 1. Routing

Gambar 5: Routing



- / → voting publik (Blade).
- /admin/\* → dashboard admin (Filament).
- /api/\* → data JSON (jika diaktifkan).

## 2. Spesifikasi Non-Fungsional

- Dijalankan di browser.
- Tidak memerlukan performa tinggi.
- Dapat digunakan di desktop & mobile.
- Sistem sederhana, single-server.



## Business Requirement Document (BRD)

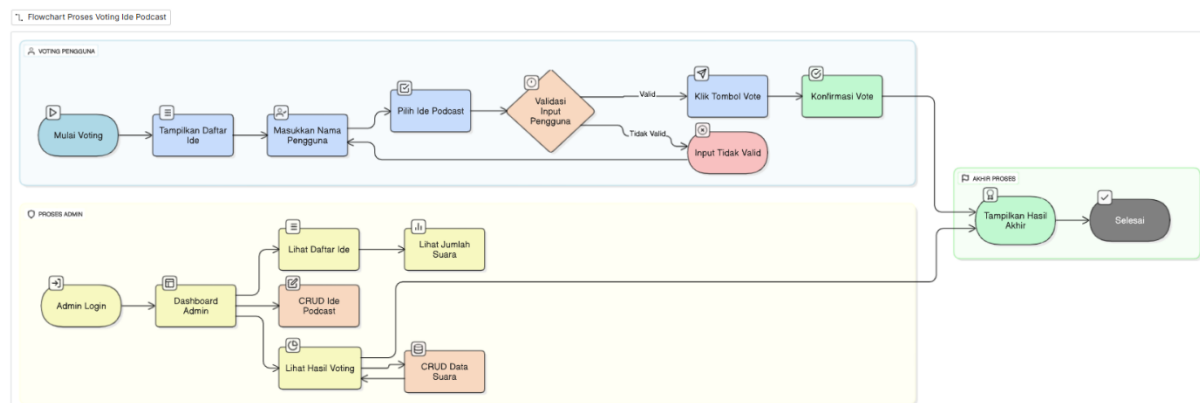
### 1. Ringkasan Eksekutif

Sistem **Voting Ide Podcast** adalah aplikasi berbasis web yang dirancang untuk memungkinkan pengguna memberikan suara pada ide podcast yang telah disediakan. Sistem ini memiliki antarmuka untuk pemilih publik dan dashboard admin untuk mengelola ide, suara, serta interaksi pengguna. Tujuan bisnis dari aplikasi ini adalah menyediakan pengalaman voting yang mudah digunakan serta memudahkan admin dalam mengelola ide-ide dan suara yang masuk.

### 2. Tujuan Bisnis

- **Tujuan Utama:** Menyediakan platform bagi pengguna untuk memberikan suara pada ide podcast, memungkinkan pencipta konten untuk mengumpulkan umpan balik audiens secara real-time.
- **Tujuan Sekunder:** Menyediakan alat bagi admin untuk mengelola ide-ide podcast, memantau tren voting, dan melacak partisipasi pengguna.

### 3. Flowchart



#### Proses Voting Pengguna:

##### 1. Mulai Voting

Pengguna mengakses halaman utama untuk memulai proses voting.

##### 2. Tampilkan Daftar Ide

Daftar ide podcast ditampilkan kepada pengguna, memungkinkan mereka untuk memilih salah satu ide yang ingin mereka beri suara.

### 3. Masukkan Nama Pengguna

Pengguna diminta untuk memasukkan nama mereka sebelum memberikan suara.

### 4. Pilih Ide Podcast

Pengguna memilih satu ide podcast dari daftar yang disediakan.

### 5. Validasi Input Pengguna

Sistem memvalidasi apakah nama pengguna telah diisi dan apakah ide yang dipilih ada dalam daftar. Ini adalah langkah penting untuk memastikan bahwa input dari pengguna valid.

- **Jika Valid:** Proses melanjutkan ke langkah berikutnya.
- **Jika Tidak Valid:** Sistem akan memberi tahu pengguna untuk memperbaiki input mereka, dan mereka diminta untuk mengisi ulang informasi yang diperlukan.

### 6. Klik Tombol Vote

Setelah memasukkan nama dan memilih ide, pengguna menekan tombol "Vote" untuk mengirimkan suara mereka.

### 7. Konfirmasi Vote

Sistem mengonfirmasi bahwa suara telah berhasil dikirimkan dan memberi pesan bahwa voting berhasil.

### 8. Selesai

Proses voting oleh pengguna selesai.

## Proses Admin:

### 1. Admin Login

Admin masuk ke dalam sistem dengan menggunakan kredensial login untuk mendapatkan akses ke dashboard admin.

### 2. Dashboard Admin

Admin dapat mengakses dashboard, yang memberikan tampilan pengelolaan ide dan voting.

### 3. CRUD Ide Podcast

Admin dapat melakukan **Create**, **Read**, **Update**, dan **Delete** pada ide-ide podcast yang ada. Hal ini memungkinkan admin untuk menambah, memperbarui, atau menghapus ide-ide yang tersedia untuk voting.

#### **4. Lihat Daftar Ide**

Admin dapat melihat daftar ide podcast yang tersedia beserta jumlah suara yang diterima oleh setiap ide.

#### **5. Lihat Jumlah Suara**

Admin dapat melihat total suara yang diterima untuk setiap ide, memberikan gambaran umum tentang bagaimana voting berjalan.

#### **6. CRUD Data Suara**

Admin dapat mengelola data suara, seperti memperbaiki atau menghapus suara jika diperlukan.

#### **7. Tampilkan Hasil Akhir**

Setelah admin mengelola dan memantau voting, hasil akhir dari proses voting ditampilkan, memberi tahu semua pihak mengenai ide yang mendapatkan suara terbanyak.

#### **8. Selesai**

Proses admin selesai setelah hasil akhir ditampilkan.

### **Ringkasan Alur:**

Flowchart ini secara keseluruhan menggambarkan dua proses utama:

- **Proses Pengguna:** Pengguna memberikan suara untuk ide podcast setelah memvalidasi input mereka.
- **Proses Admin:** Admin mengelola ide dan voting serta menampilkan hasil akhir.

#### **4. Lingkup Pekerjaan**

- **Antarmuka Publik:** Pengguna dapat memberikan suara pada ide podcast dengan memasukkan nama dan memilih ide. Setiap pengguna hanya bisa memberikan satu suara per ide untuk memastikan keadilan dan hasil voting yang akurat.
- **Antarmuka Admin:** Admin dapat:
  - Melihat dan mengelola ide-ide podcast (fungsi CRUD)
  - Memantau suara, termasuk jumlah suara per ide dan informasi detail pemilih
  - Mengekspor data atau melihat hasil melalui dashboard.

Platform ini akan diimplementasikan menggunakan **Laravel** dan **Filament** untuk fungsionalitas backend dan admin, serta template **Blade** untuk tampilan frontend.

## 5. Pemangku Kepentingan (*Stakeholders*)

- **Pengguna Akhir:** Publik yang akan memberikan suara pada ide (tanpa login).
- **Pengguna Admin:** Individu yang memiliki hak akses admin untuk mengelola ide dan proses voting (dengan login).

## 6. Kebutuhan

### Kebutuhan Fungsional

- **Fungsi Voting:**
  - Pengguna dapat memberikan suara dengan memasukkan nama dan memilih ide.
  - Setiap pengguna hanya bisa memberikan satu suara per ide.
  - Sistem harus mengonfirmasi pengiriman suara.
- **Dashboard Admin:**
  - Admin dapat melihat semua ide dan jumlah suara masing-masing ide.
  - Admin dapat melakukan operasi CRUD pada ide.
  - Admin dapat melihat data voting secara detail, termasuk nama pemilih dan ide yang dipilih.
- **Integrasi API:** Menyediakan endpoint API untuk mendapatkan daftar ide dan suara terbanyak dalam format JSON (GET/POST).

### Kebutuhan Non-Fungsional

- **Keamanan:**
  - Menggunakan autentikasi berbasis peran (RBAC) untuk mengontrol akses, memastikan hanya admin yang dapat mengelola ide dan suara.
- **Kegunaan:**
  - Antarmuka pengguna yang sederhana dan intuitif untuk pengguna dan admin.
- **Performa:**
  - Sistem harus berjalan dengan efisien di semua browser modern tanpa membutuhkan performa tinggi.

- **Ketersediaan:**
  - Sistem harus dapat diakses 24/7 untuk pengguna melakukan voting dan untuk admin mengelola data.

## 7. Target Audiens

- **Pencipta Podcast:** Untuk mengumpulkan umpan balik dan suara tentang ide podcast mereka.
- **Audiens Umum:** Pengguna yang tertarik untuk memberikan suara pada ide podcast favorit mereka.

## 8. Batasan

- **Satu Suara Per Pengguna:** Sistem hanya akan memungkinkan satu suara per pengguna untuk setiap ide untuk mencegah manipulasi suara.
- **Tanpa Pendaftaran Pengguna:** Pengguna tidak perlu membuat akun atau login untuk memberikan suara.

## 9. Analisis Pasar

- **Target Pasar:** Individu yang terlibat dalam pembuatan konten atau podcast yang ingin mengumpulkan umpan balik audiens.
- **Kebutuhan Pasar:** Aplikasi ini akan melayani pencipta podcast dan audiens mereka yang mencari mekanisme umpan balik yang lebih interaktif dan langsung.

## 10. Kriteria Keberhasilan

- Implementasi yang sukses dari fungsi voting dan operasi CRUD pada ide dan jumlah suara.
- Pengelolaan dan pemantauan yang lancar oleh admin melalui dashboard admin.