**Nama: Ahmad Karta Nugraha**

**NPM: 714230035**

# 1. Nama Aplikasi

Aplikasi Pemantauan Emosi Harian

# 2. Deskripsi Aplikasi

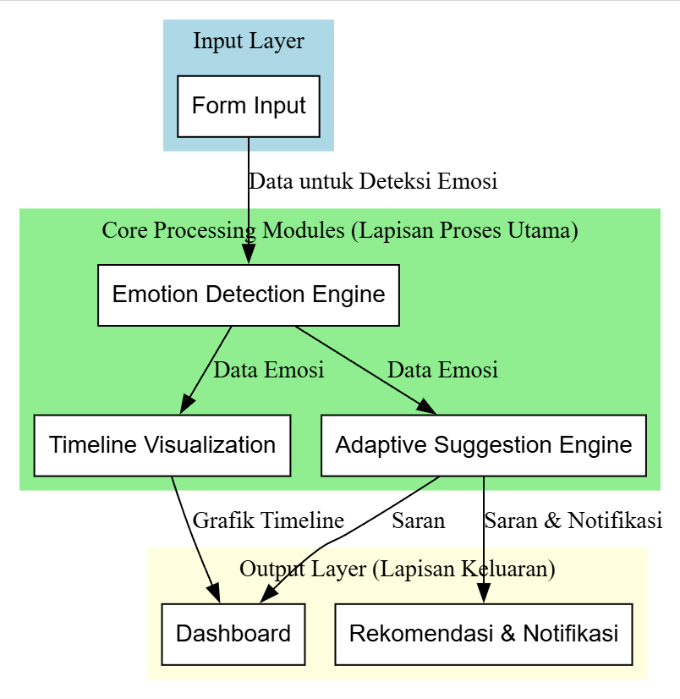
Aplikasi Pemantauan Emosi Harian adalah platform yang dirancang untuk membantu pengguna dalam mengenali, memantau, dan mengelola emosi mereka sehari-hari. Aplikasi ini memungkinkan pengguna untuk mencatat emosi mereka secara manual, menganalisis pola emosi yang muncul, serta menerima rekomendasi untuk meningkatkan kesejahteraan emosional mereka. Dengan menggunakan fitur-fitur seperti grafik tren emosi dan pemberitahuan pengingat, Aplikasi ini dirancang untuk membantu pengguna lebih memahami perasaan mereka dan memberikan dorongan agar mereka bisa menjaga keseimbangan emosional dengan lebih baik.

Aplikasi ini dapat digunakan oleh individu yang ingin lebih sadar akan kondisi emosional mereka, baik untuk refleksi pribadi maupun untuk meningkatkan kesejahteraan mental mereka. Dengan menyediakan data analitik, aplikasi ini juga memberikan wawasan tentang pola emosi pengguna dari waktu ke waktu dan memberikan saran yang dapat membantu mereka mengelola emosi negatif, seperti stres atau kecemasan.

# 3. Daftar fitur

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Fitur** | **Deskripsi Singkat** |
| 1 | Pencatatan Emosi | Pengguna bisa memilih salah satu emosi dari pilihan yang ada, seperti senang, cemas, marah, atau sedih, atau cukup menuliskan perasaan mereka secara singkat. |
| 2 | Grafik Tren Emosi | |  | | --- | | Menampilkan grafik yang menggambarkan perubahan emosi pengguna dari waktu ke waktu. Grafik ini bisa berupa grafik garis atau batang yang menunjukkan perbandingan antara emosi positif dan negatif selama periode tertentu (misalnya, mingguan atau bulanan). |  |  | | --- | |  | |
| 3 | |  | | --- | | Rekomendasi Aktivitas |  |  | | --- | |  | | |  | | --- | | Berdasarkan pola emosi yang terdeteksi, aplikasi memberikan rekomendasi aktivitas yang dapat membantu meningkatkan suasana hati pengguna, seperti meditasi, olahraga, atau bahkan membaca. Jika aplikasi mendeteksi stres atau kecemasan, saran untuk relaksasi atau latihan pernapasan akan diberikan. |  |  | | --- | |  | |

# 4. Arsitektur Sistem



**Input Layer (Lapisan Input)**  
Ini adalah bagian di mana aplikasi menerima data dari pengguna. Di sini, data dimasukkan melalui **Form Input**. Data ini bisa berupa informasi yang diisi oleh pengguna tentang perasaan atau kondisi emosional mereka.

**Core Processing Modules (Lapisan Proses Utama)**  
Setelah data masuk dari form, data tersebut diproses di bagian inti sistem, yang terdiri dari:

* **Emotion Detection Engine**: Bagian ini bertugas membaca dan menganalisis data dari input untuk mendeteksi jenis emosi yang sedang dialami oleh pengguna.
* **Timeline Visualization**: Mengubah hasil analisis emosi menjadi grafik timeline yang menunjukkan bagaimana emosi pengguna berubah selama waktu tertentu.
* **Adaptive Suggestion Engine**: Berdasarkan emosi yang terdeteksi, bagian ini memberikan saran yang sesuai, seperti rekomendasi untuk istirahat atau latihan relaksasi.

**Output Layer (Lapisan Keluaran)**  
Setelah proses analisis dan pengolahan selesai, hasilnya ditampilkan kepada pengguna dalam dua bentuk utama:

* **Dashboard**: Tempat untuk melihat grafik timeline dan laporan emosi secara visual, memudahkan pengguna dan fasilitator memahami dinamika emosi selama sesi.
* **Rekomendasi & Notifikasi**: Saran atau pemberitahuan langsung yang diberikan kepada pengguna untuk membantu mereka mengelola emosi secara lebih baik.

# 5. Tabel Fitur, Atribut, method dan class

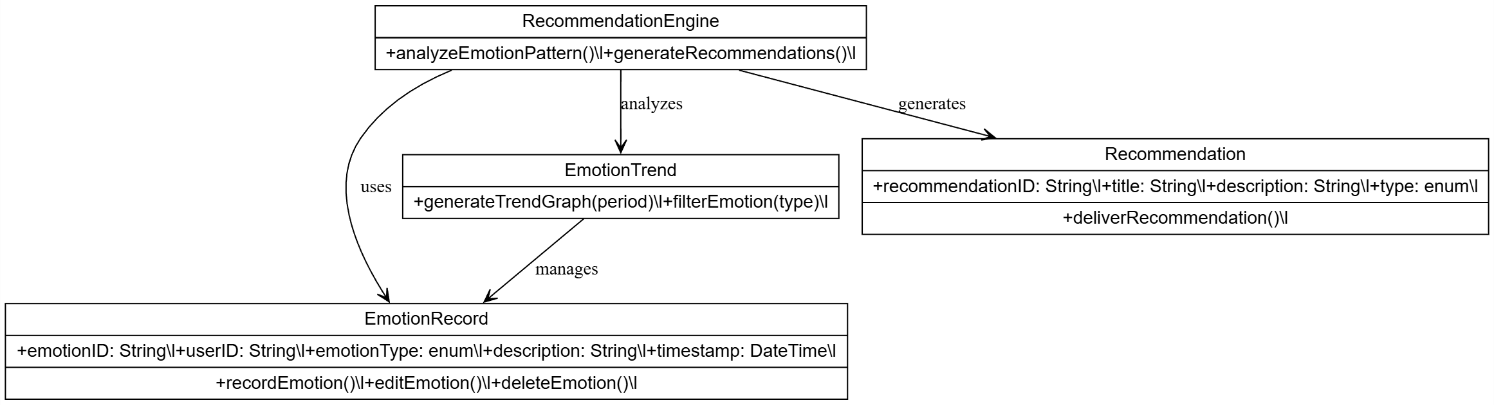
| **No** | **Nama Fitur** | **Atribut (Tipe Data)** | **Method** | **Nama Class** |
| --- | --- | --- | --- | --- |
| 1 | Pencatatan Emosi | - emotionID: String- userID: String- emotionType: enum {Senang, Cemas, Marah, Sedih, Lainnya}- description: String- timestamp: DateTime | - recordEmotion(): void- editEmotion(): void- deleteEmotion(): void | EmotionRecord |
| 2 | Grafik Tren Emosi | - userID: String- emotionHistory: List- startDate: Date- endDate: Date | - generateTrendGraph(period: String): GraphObject- filterEmotion(emotionType): List | EmotionTrend |
| 3 | Rekomendasi Aktivitas | - userID: String- detectedPattern: String- recommendationList: List- lastEmotionRecord: EmotionRecord | - analyzeEmotionPattern(): void- generateRecommendations(): List | RecommendationEngine |
|  |  |  |  | Recommendation |
|  |  | - recommendationID: String- title: String- description: String- type: enum {Meditasi, Olahraga, Membaca, Relaksasi, Pernapasan} | - deliverRecommendation(): void |  |

Penjelasan :

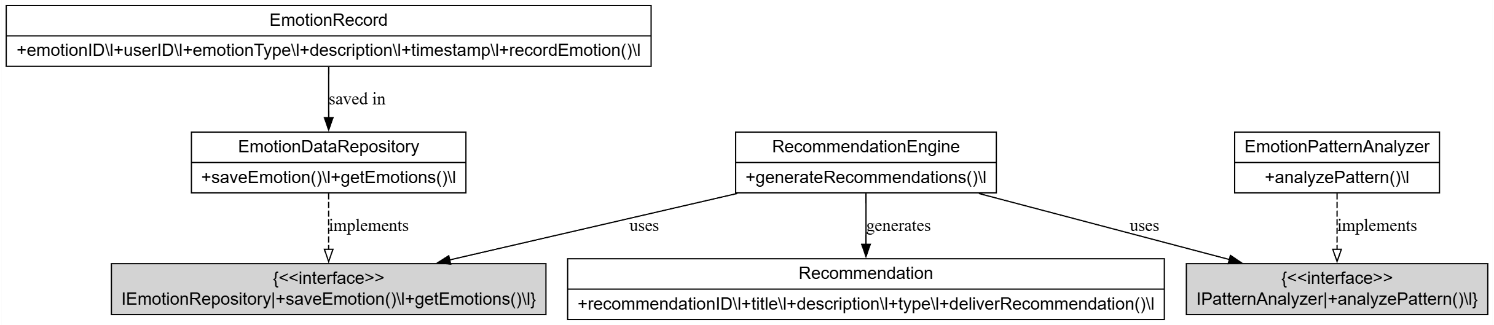
* **EmotionRecord  
  Menangani pencatatan emosi harian yang dimasukkan oleh pengguna, termasuk tipe emosi dan deskripsi singkat.**
* **EmotionTrend  
  Bertugas mengelola dan membuat grafik tren emosi pengguna dari data rekaman emosi dalam rentang waktu tertentu.**
* **RecommendationEngine  
  Modul untuk menganalisis pola emosi pengguna dan menghasilkan daftar rekomendasi aktivitas yang sesuai untuk membantu pengguna mengelola emosi mereka.**
* **Recommendation  
  Kelas yang merepresentasikan masing-masing rekomendasi aktivitas, termasuk tipe dan deskripsi.**

# 6. Class Diagram

Sebelum Refactor:

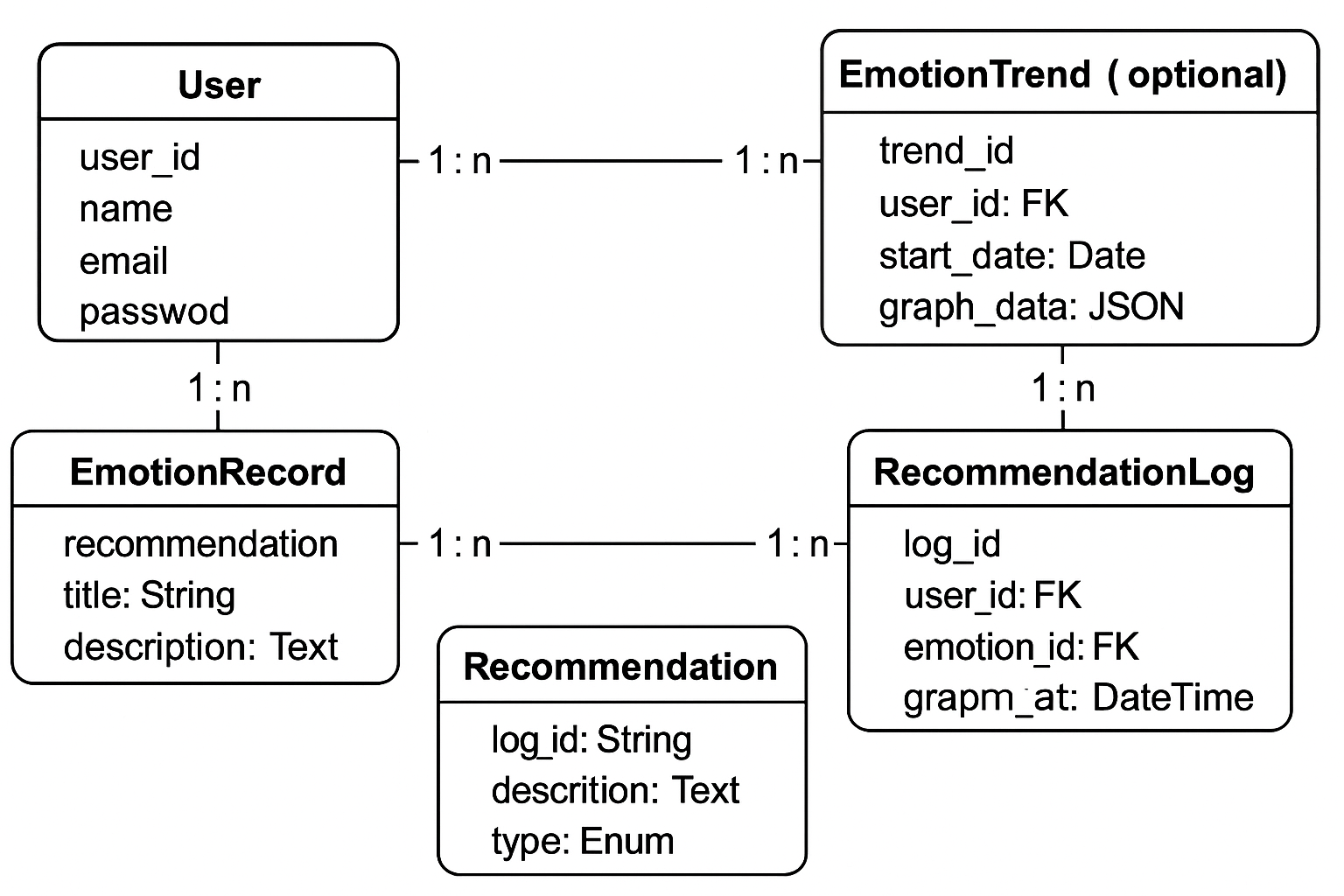


Setelah Refactor:



| **No** | **Prinsip / Aturan** | **Pelanggaran “Sebelum”** |
| --- | --- | --- |
| 1 | **SRP (Single Responsibility Principle)** | Satu kelas (misal: RecommendationEngine) mengurus banyak hal sekaligus: menyimpan data, menganalisis, dan membuat rekomendasi. |
| 2 | **OCP (Open/Closed Principle)** | Saat ingin menambahkan jenis pola analisis atau jenis rekomendasi, harus mengubah kode di kelas utama (RecommendationEngine). |
| 3 | **LSP (Liskov Substitution Principle)** | Tidak ada interface abstrak, sehingga jika membuat subclass/versi lain dari data atau rekomendasi, sistem tidak stabil. |
| 4 | **ISP (Interface Segregation Principle)** | Tidak ada pemisahan antara modul penyimpanan dan analisis; semua langsung saling terhubung dalam satu class besar. |
| 5 | **DIP (Dependency Inversion Principle)** | Kelas RecommendationEngine langsung bergantung pada implementasi konkret EmotionDataRepository dan EmotionPatternAnalyzer, bukan pada abstraksi/interface. |

# 7. ERD



**Relasi Antar Entitas**

1. **User ↔ EmotionRecord**
   * **Relasi:** 1 user dapat memiliki banyak catatan emosi (*1 : n*).
   * **Makna:** Setiap pengguna bisa mencatat emosinya setiap hari.
2. **User ↔ EmotionTrend** *(opsional)*
   * **Relasi:** 1 user memiliki banyak data tren emosi (*1 : n*).
   * **Makna:** Tren dibuat berdasarkan catatan emosi berkala (mingguan/bulanan).
3. **User ↔ RecommendationLog**
   * **Relasi:** 1 user bisa menerima banyak rekomendasi (*1 : n*).
   * **Makna:** Pengguna mendapat saran berdasarkan emosi yang dicatat.
4. **EmotionRecord ↔ RecommendationLog**
   * **Relasi:** 1 catatan emosi dapat menghasilkan banyak log rekomendasi (*1 : n*).
   * **Makna:** Dari satu emosi, bisa muncul lebih dari satu saran.
5. **Recommendation ↔ RecommendationLog**
   * **Relasi:** 1 jenis rekomendasi dapat digunakan oleh banyak pengguna (*1 : n*).
   * **Makna:** Satu saran (misalnya: meditasi) bisa dikaitkan ke berbagai emosi pengguna.

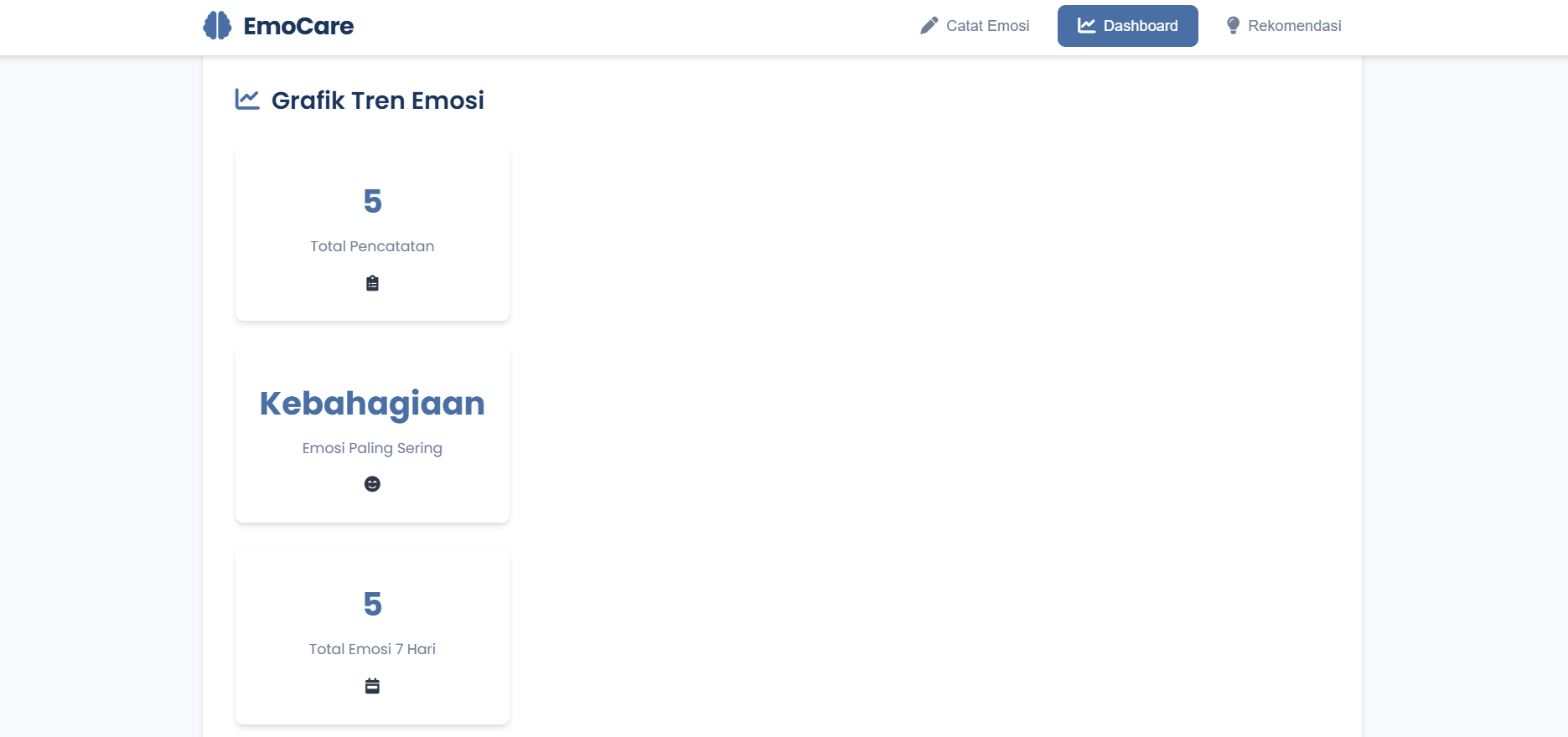
**Alur Penggunaan Singkat**

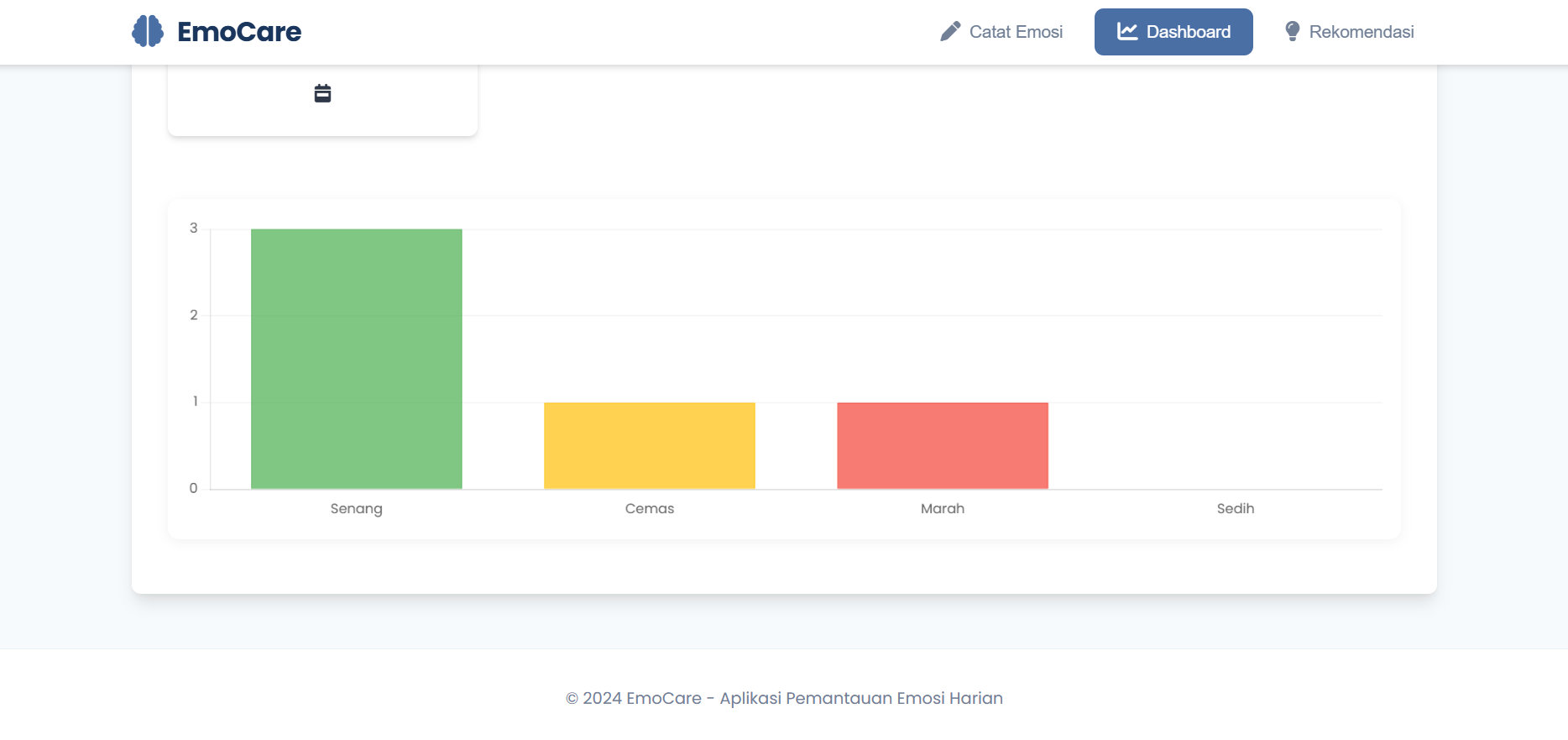
1. **Pengguna login** (data di entitas User).
2. **Mengisi form emosi** → disimpan sebagai EmotionRecord.
3. Sistem **menganalisis pola emosi** → memunculkan Recommendation.
4. Rekomendasi tersebut **dicatat dalam RecommendationLog** untuk histori.
5. (Opsional) Sistem menghasilkan **EmotionTrend** berdasarkan histori untuk visualisasi grafik.

# 8. Daftar antarmuka

1. **Dashboard**

Menampilkan grafik batang jumlah emosi dan Menampilkan informasi ringkas seperti total pencatatan, emosi yang paling sering muncul, dan total emosi 7 hari terakhir. yang dicatat pengguna selama periode tertentu berdasarkan emosi yang telah pengguna catat sebelumnya.

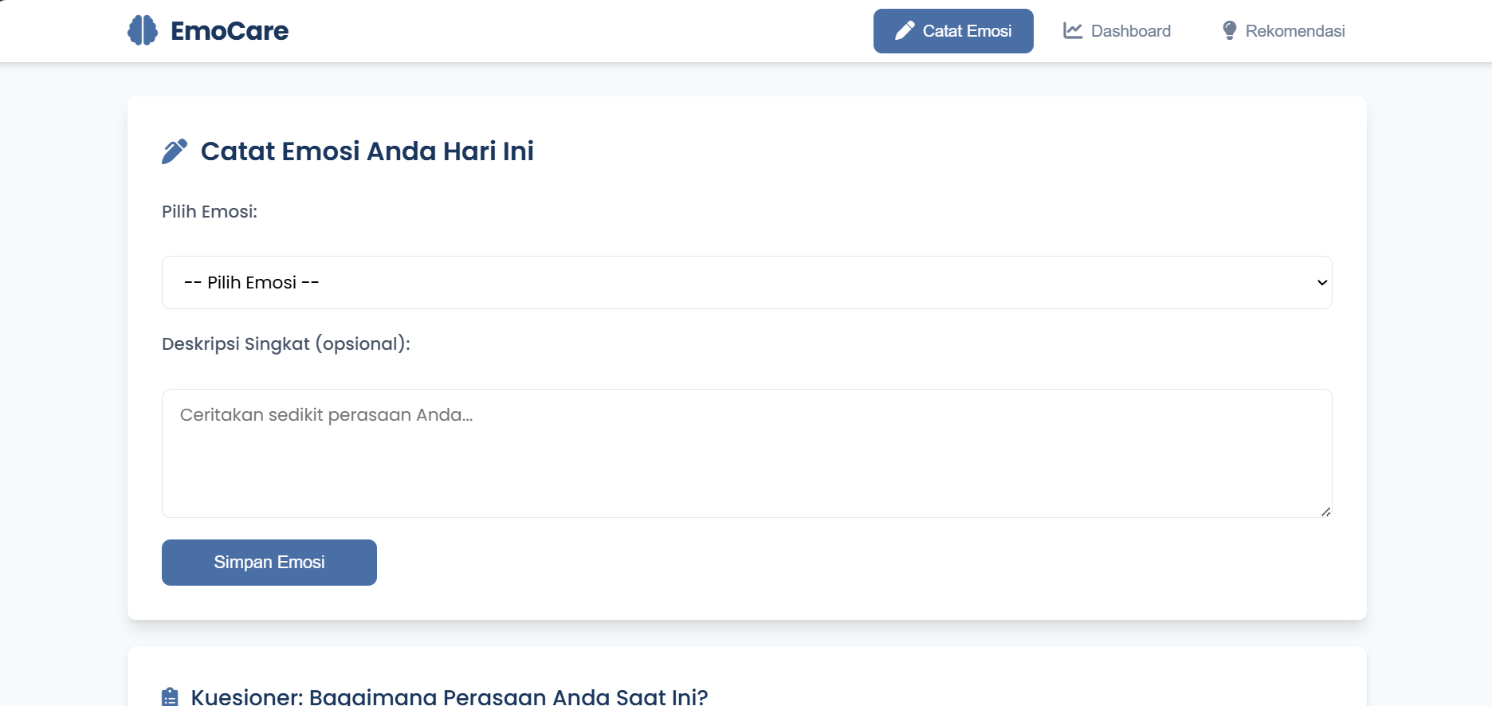




1. **Pencatat Emosi**

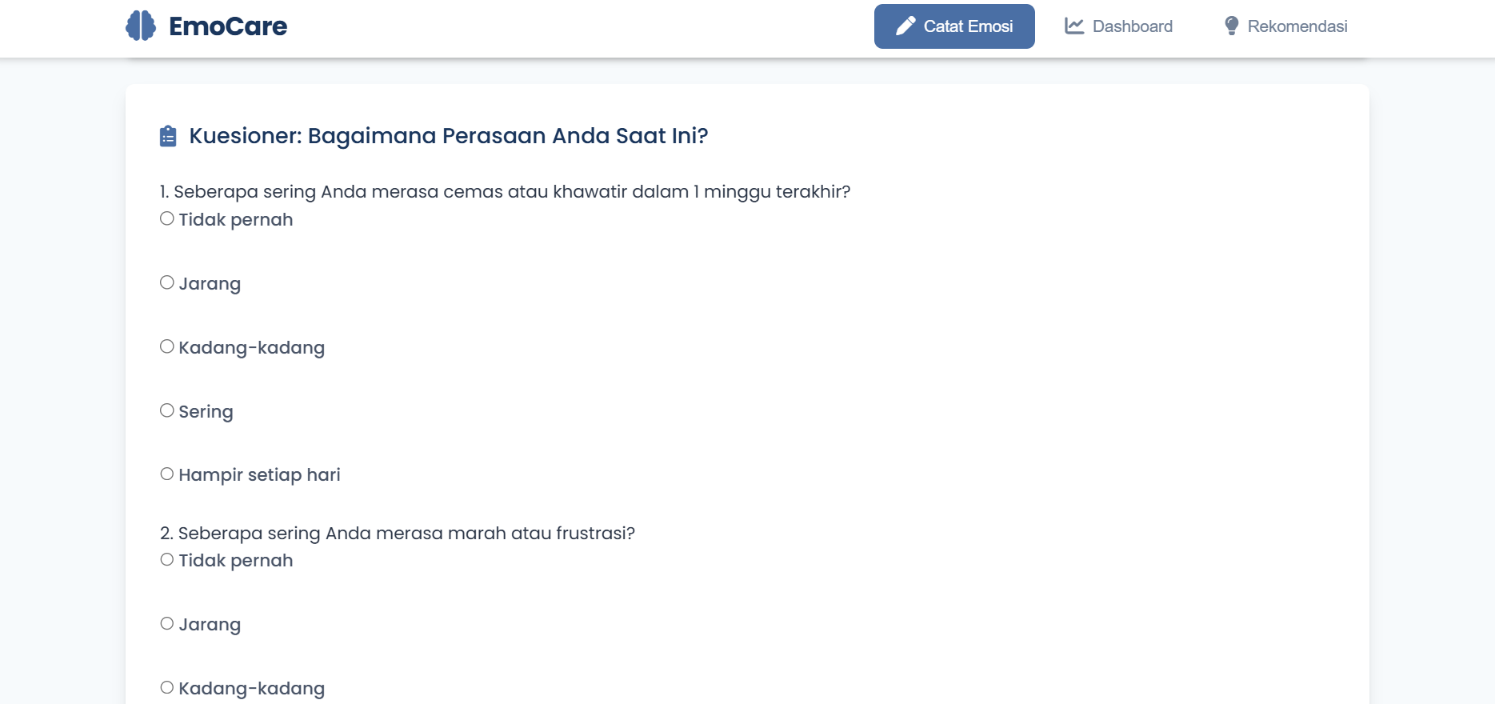
* **Mencatat Emosi**

Pengguna dapat memilih jenis emosi yang sedang dirasakan dan menambahkan deskripsi singkat secara manual.



* **Quisioner**

Pengguna menjawab beberapa pertanyaan tentang frekuensi emosi negatif seperti kecemasan atau kemarahan.



1. **Rekomendasi aktifitas**

Menampilkan rekomendasi aktivitas yang sesuai dengan kondisi emosional terakhir pengguna (baik dari input manual maupun hasil kuesioner).

