

LAPORAN PROYEK MATA KULIAH TEKNOLOGI BASIS DATA
SEMESTER GENAP 2024/2025
MY JOURNAL



t35410n1k4t3n94h

Anggota Tim

1. 23/518300/TK/57035 | Fanny Elisabeth Panjaitan

Departemen Teknik Elektro dan Teknologi Informasi
Fakultas Teknik
Universitas Gadjah Mada
2025

1. DESKRIPSI APLIKASI

Dalam kehidupan sehari-hari, banyak individu menghadapi tantangan dalam mengelola waktu, menyusun prioritas, serta menjaga kesehatan mental melalui refleksi diri. Di tengah padatnya aktivitas dan informasi yang terus-menerus datang, kebutuhan akan sebuah platform digital yang dapat membantu seseorang mencatat pengalaman, melacak suasana hati, dan mengatur tugas harian menjadi semakin penting. Seperti seorang mahasiswa tingkat akhir mengalami kesulitan dalam membagi waktu antara tugas akhir, kegiatan organisasi, dan pekerjaan paruh waktu. Ia mencoba menggunakan aplikasi to-do list, tetapi merasa kurang bisa mengekspresikan pikiran dan perasaannya secara bebas. Di sisi lain, aplikasi journaling yang ia coba tidak terintegrasi dengan fitur manajemen tugas, sehingga membuatnya harus berpindah-pindah platform. Hal ini tidak hanya mengurangi produktivitas, tetapi juga membuat proses refleksi diri menjadi terfragmentasi.

Situasi serupa juga terjadi pada para pekerja remote yang perlu mencatat progres kerja dan kondisi emosional harian, namun kesulitan mencari platform yang mendukung kedua kebutuhan secara terpadu dan privat. Melihat adanya celah ini, dibutuhkan solusi digital yang tidak hanya fokus pada pencatatan tugas atau jurnal semata, tetapi mengintegrasikan keduanya dalam satu aplikasi yang aman, personal, dan interaktif. Inilah yang mendorong lahirnya aplikasi ini.

Aplikasi ini merupakan aplikasi web *journaling digital* yang lengkap (*full-featured*), dibangun menggunakan React (Next.js) dan Supabase, yang dirancang untuk membantu pengguna mencatat, mengelola, dan merefleksikan pengalaman harian mereka secara terstruktur dan interaktif. Aplikasi ini mendukung sistem autentikasi pengguna (login, register, logout), memastikan bahwa setiap data pengguna tersimpan secara aman dan privat dalam database PostgreSQL yang dikelola oleh Supabase.

Pengguna dapat membuat, mengedit, dan menghapus *journal entries* yang dapat diberi *tag* (label) dan dikaitkan dengan *mood* yang dipilih dari kumpulan emoji yang tersedia. Antarmuka aplikasi mengusung gaya *glassmorphism modern*, dengan desain responsif dan animasi halus yang diimplementasikan menggunakan Tailwind CSS serta kelas utilitas kustom.

Beberapa fitur utama aplikasi ini adalah sebagai berikut.

- Dashboard Statistik Real-Time, menampilkan informasi seperti jumlah total entri, jumlah kata, jumlah entri favorit, dan statistik penggunaan tag.

- Mood Tracker dan Streak Tracking, yang dirancang untuk mendorong konsistensi dalam menulis jurnal harian.
- Sistem Filter & Pencarian yang fleksibel, memungkinkan pengguna menyaring entri berdasarkan kata kunci, tag, mood, rentang tanggal, dan entri favorit.
- Fitur Favorite, untuk menandai entri penting agar mudah diakses kembali.
- Sistem Tagging yang cerdas, mendukung saran tag dan mencegah duplikasi atau penggunaan tag yang berlebihan.

Untuk mendukung produktivitas, aplikasi ini juga menyediakan fitur sebagai berikut.

- Catatan Cepat (Quick Notes) dan To-Do List, yang mendukung fungsi CRUD (Create, Read, Update, Delete).
- To-do list dilengkapi dengan pelacakan tenggat waktu, persentase penyelesaian, dan indikator visual progres.

Selain itu, pengguna dapat mengekspor data jurnal mereka dalam berbagai format seperti JSON, TXT, Markdown, CSV, dan PDF, dengan opsi untuk menyertakan tag, mood, dan statistik, sehingga memudahkan proses *backup* maupun berbagi.

Dari sisi arsitektur yang menggunakan arsitektur 2-tier (Client ↔ Server/Database), *frontend* disusun menggunakan komponen yang dapat digunakan kembali (modals, cards, filter, statistik), untuk menjaga *scalability* dan kemudahan pemeliharaan. Semua operasi data (CRUD untuk jurnal, tag, to-do, dan catatan) dijalankan melalui Supabase RESTful API, dengan pembaruan UI secara *optimistic* dan sistem penanganan error yang memastikan pengalaman pengguna tetap mulus. Aplikasi ini juga telah dioptimalkan untuk aksesibilitas dan penggunaan lintas perangkat, menjadikannya solusi digital yang modern, andal, dan berorientasi pada pengguna untuk keperluan journaling dan manajemen produktivitas harian.

2. ANALISIS KEBUTUHAN

Analisis kebutuhan bertujuan untuk mengidentifikasi dan mendefinisikan fitur serta spesifikasi teknis yang harus dipenuhi oleh sistem agar dapat bekerja sesuai dengan tujuan. Aplikasi My Journal ini dirancang untuk memberikan solusi digital terpadu dalam mencatat jurnal harian serta mengelola tugas, dengan pendekatan berbasis web, antarmuka modern, dan sistem backend yang efisien serta aman.

2.1. Kebutuhan Fungsional

Kebutuhan fungsional menggambarkan fitur inti yang wajib dimiliki oleh sistem agar aplikasi dapat berjalan sesuai dengan tujuan dan memenuhi ekspektasi pengguna. Dalam konteks aplikasi Journaling & To-Do List, kebutuhan ini mencakup seluruh tindakan atau proses yang dapat dilakukan oleh pengguna melalui antarmuka aplikasi (UI), serta logika sistem di belakang layar (backend logic) yang mendukung kelancaran operasional aplikasi. Berikut adalah daftar kebutuhan fungsional utama yang telah diidentifikasi dan dijadikan dasar dalam perancangan serta pengembangan aplikasi.

Tabel 1 Kebutuhan Fungsional

Req-ID	Kebutuhan Fungsional	Deskripsi
F-01	Aplikasi berbasis web	Aplikasi dikembangkan sebagai web app responsif menggunakan framework Next.js (berbasis React). Hal ini memungkinkan aplikasi dapat berjalan di browser tanpa perlu instalasi tambahan, serta mendukung pengembangan single-page application (SPA) dengan performa yang optimal.
F-02	Menggunakan DBMS Relasional	Sistem basis data menggunakan Supabase yang berbasis PostgreSQL, mendukung relasi kompleks antar tabel, serta memiliki RESTful API bawaan untuk operasi data. Supabase juga menyediakan layanan autentikasi dan penyimpanan yang terintegrasi.
F-03	Minimal 3 Tabel dan Relasi Many-to-Many	Struktur database mencakup journals yang menyimpan entri jurnal, tags yang menyimpan daftar tag yang dapat digunakan oleh pengguna, dan journal_tags yang merupakan tabel relasi many-to-many yang menghubungkan jurnal dengan banyak tag. Relasi ini memungkinkan satu entri jurnal memiliki banyak tag, dan satu tag dapat digunakan di banyak entri jurnal.
F-04	CRUD Operations untuk Journals dan Todos	Aplikasi mendukung operasi Create, Read, Update, Delete (CRUD) penuh terhadap entri jurnal dan to-do list. Operasi ini dapat diakses melalui UI yang intuitif, dan setiap perubahan tercermin langsung melalui integrasi dengan API Supabase.
F-05	Konsep VIEW: journal_entries_with_tags	Untuk memudahkan tampilan data gabungan antara jurnal dan tag, digunakan pendekatan VIEW (secara konsep) bernama journal_entries_with_tags, baik secara manual pada client maupun melalui join query di

		Supabase. Hal ini memungkinkan pengguna melihat tag terkait langsung dalam entri jurnal tanpa harus melakukan pencarian manual ke tabel relasi.
F-06	Fitur Ekspor Data	Pengguna dapat mengekspor data jurnal mereka ke dalam format JSON dan CSV, baik untuk tujuan backup maupun analisis pribadi. Proses ekspor mencakup data jurnal, tag, dan metadata lainnya yang relevan, dan dapat dipilih berdasarkan filter tertentu.
F-07	To-Do List dengan CRUD & Tracking	Fitur to-do list mendukung penambahan, pengeditan, dan penghapusan tugas, serta pelacakan status penyelesaian (selesai/belum). Sistem juga menghitung progres penyelesaian dan dapat memfilter tugas berdasarkan tenggat waktu.
F-08	Mood Tracker dan Favorite	Setiap entri jurnal dapat dikaitkan dengan suasana hati (mood) yang dipilih dari daftar emoji. Pengguna juga dapat menandai entri sebagai <i>favorite</i> untuk akses cepat.
F-09	Filtering & Search	Sistem pencarian dan filter mendukung pencarian berdasarkan kata kunci, tag, tanggal, mood, dan status favorit. Hal ini memungkinkan pengguna menemukan catatan spesifik dengan mudah.
F-10	Statistik Jurnal	Dashboard menampilkan statistik seperti jumlah total entri, jumlah kata, distribusi mood, jumlah entri favorit, serta penggunaan tag terbanyak. Ini mendorong kebiasaan menulis yang konsisten dan teratur.

2.2. Kebutuhan Non Fungsional

Kebutuhan non-fungsional menjelaskan aspek-aspek pendukung dalam pengembangan aplikasi yang tidak berkaitan langsung dengan fitur utama, namun mempengaruhi kualitas sistem secara keseluruhan seperti teknologi yang digunakan, keterbatasan sumber daya, serta struktur kerja tim. Berikut adalah rincian kebutuhan non-fungsional dari aplikasi My Journal.

2.2.1. Pilihan Teknologi dan Tools

Aplikasi ini dibangun menggunakan teknologi modern yang mendukung pengembangan web secara cepat, fleksibel, dan efisien. Untuk bagian frontend, digunakan Next.js, yaitu framework berbasis React yang mendukung server-side rendering dan static site generation. Pemilihan Next.js memberikan keuntungan dalam hal kecepatan akses dan struktur proyek yang rapi. Untuk styling dan desain antarmuka, digunakan Tailwind CSS, sebuah framework

utility-first yang memungkinkan pembuatan tampilan responsif dan konsisten dengan lebih sedikit kode CSS manual. Di sisi backend dan database, digunakan Supabase, sebuah platform Backend-as-a-Service (BaaS) yang menyediakan database relasional PostgreSQL, autentikasi pengguna, serta API RESTful yang otomatis untuk operasi CRUD. Supabase sangat cocok untuk proyek individu karena mengurangi beban pengelolaan server dan backend secara manual. Tools tambahan seperti Git dan GitHub digunakan untuk version control, serta Vercel digunakan sebagai platform deployment Next.js secara gratis dan efisien.

2.2.2. Keterbatasan Sumber Daya dan Waktu

Proyek ini dikembangkan oleh satu orang dengan menggunakan laptop processor Intel (R) Celeron(R) N4020 CPU @ 1.10GHz, 1101 Mhz, 2 Core(s), 2 Logical Processor(s) RAM 4 GB, sehingga terdapat keterbatasan dalam hal sumber daya manusia, alat yang digunakan, waktu pengerjaan, dan kapasitas pengujian. Seluruh proses mulai dari analisis kebutuhan, perancangan sistem, implementasi kode, hingga dokumentasi dilakukan secara mandiri. Karena itu, pengembangan aplikasi difokuskan pada fitur-fitur utama yang memberikan nilai fungsional paling tinggi, seperti sistem jurnal, tagging, to-do list, dan ekspor data. Beberapa fitur lanjutan seperti kolaborasi jurnal, mode gelap, dan integrasi AI untuk ringkasan jurnal direncanakan untuk dikembangkan pada tahap selanjutnya. Dari sisi waktu, proyek ini diselesaikan dalam rentang waktu akademik yang terbatas, sehingga setiap fitur dirancang dan diuji dengan efisiensi tinggi, tanpa mengorbankan stabilitas aplikasi. Pengujian dilakukan pada perangkat laptop yang sama, tanpa penggunaan server fisik atau cloud berbayar, demi menyesuaikan dengan keterbatasan alat yang tersedia.

2.2.3. Anggota Tim dan Pembagian Kerja

Aplikasi ini merupakan hasil pengembangan individu, sehingga seluruh aspek teknis dan non-teknis ditangani oleh satu orang yang berperan sebagai full-stack developer. Peran ini mencakup pengembangan frontend (pembuatan tampilan dan interaksi menggunakan Next.js dan Tailwind CSS), pengelolaan backend (penyusunan tabel dan relasi di Supabase, serta pengaturan autentikasi), perancangan UI/UX (pemilihan warna, tata letak, dan pengalaman pengguna), pengujian aplikasi (manual testing terhadap setiap fitur), serta penyusunan dokumentasi teknis.

2.2.4. Skalabilitas dan Maintainability

Aplikasi dirancang dengan memperhatikan aspek skalabilitas dan kemudahan pemeliharaan ke depan. Struktur kode dibangun dengan pendekatan komponen modular,

terutama pada bagian UI seperti form, kartu, modal, dan sistem filter. Hal ini memungkinkan penambahan fitur baru seperti ekspor PDF, integrasi kalender, atau statistik lanjutan tanpa harus mengubah banyak bagian dari kode yang sudah ada. Relasi data juga telah dirancang fleksibel dengan pendekatan many-to-many antara jurnal dan tag, sehingga pengguna bisa mengkustomisasi isi catatan dengan cara yang lebih terstruktur. Selain itu, karena Supabase memberikan API otomatis untuk setiap tabel, proses pengembangan dapat terus dilanjutkan tanpa perlu membangun backend manual, yang berarti aplikasi sangat mungkin untuk dikembangkan lebih lanjut dengan beban pengelolaan yang rendah.

3. PEMODELAN DAN PERANCANGAN APLIKASI

Pada tahap ini, dilakukan proses pemodelan dan perancangan sistem untuk memastikan bahwa aplikasi dikembangkan dengan struktur yang jelas, efisien, dan sesuai dengan kebutuhan pengguna. Perancangan mencakup bagaimana komponen frontend dan backend saling berinteraksi, bagaimana basis data dirancang untuk menyimpan informasi secara terstruktur, serta bagaimana antarmuka pengguna dibangun untuk memberikan pengalaman yang optimal.

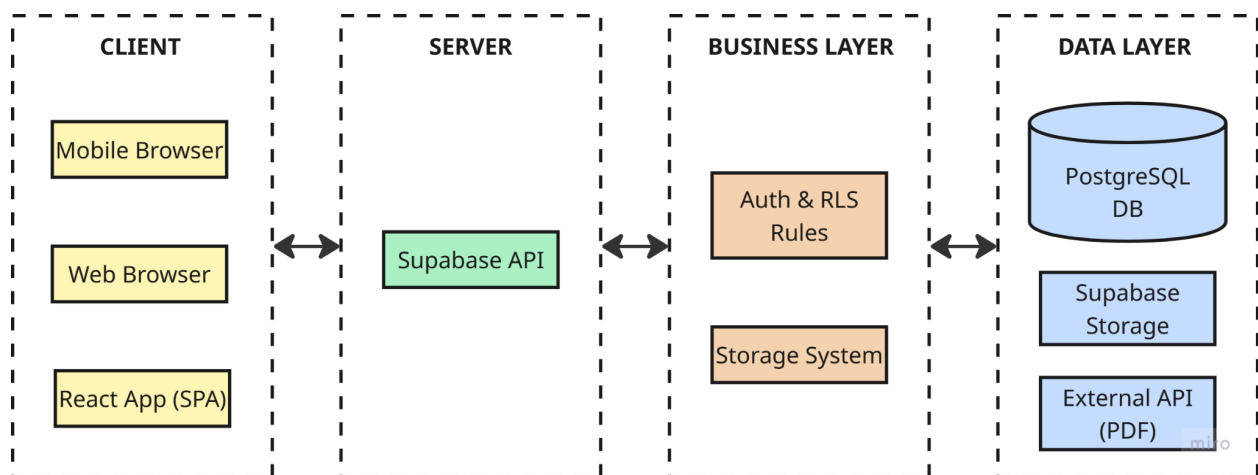
Aplikasi digital journaling ini memanfaatkan arsitektur 2-tier, dengan frontend yang dikembangkan menggunakan Next.js dan backend yang dikelola oleh Supabase, sebuah layanan Backend-as-a-Service berbasis PostgreSQL. Pemodelan data dilakukan dengan memperhatikan hubungan antar entitas penting seperti jurnal, tag, mood, todo list, dan statistik pengguna. Sementara itu, desain UI mengutamakan keterbacaan, keindahan visual melalui efek glassmorphism, dan fungsionalitas responsif untuk berbagai perangkat. Langkah ini menjadi fondasi penting dalam implementasi aplikasi, agar pengembangan sistem berjalan terarah, scalable, dan mudah dipelihara di masa mendatang.

3.1. Perancangan Arsitektur Aplikasi

Aplikasi ini menggunakan arsitektur 2-tier secara implementasi, di mana frontend berinteraksi langsung dengan backend tanpa adanya server middleware tambahan seperti Express.js atau Node.js. Pada sisi frontend, aplikasi dibangun menggunakan Next.js, yang dikombinasikan dengan Tailwind CSS untuk menghasilkan antarmuka pengguna yang modern, responsif, dan mendukung gaya visual glassmorphism. Aplikasi ini mendukung berbagai perangkat, dengan komponen-komponen antarmuka yang reusable dan tersusun modular untuk memudahkan pengembangan dan pemeliharaan.

Pada sisi backend, aplikasi sepenuhnya mengandalkan Supabase, yang menyediakan berbagai layanan penting seperti RESTful API, otentikasi pengguna, dan database relasional PostgreSQL. Supabase juga mendukung fitur-fitur penting seperti Row-Level Security (RLS) untuk memastikan setiap pengguna hanya dapat mengakses datanya sendiri, serta penyimpanan file melalui Supabase Storage. Semua operasi CRUD untuk entri jurnal, tag, mood, catatan cepat, dan todo list dilakukan langsung melalui Supabase, baik melalui REST API maupun Supabase SDK yang terintegrasi dengan frontend.

Meski secara implementasi arsitektur ini bersifat 2-tier karena frontend langsung terhubung ke layanan Supabase, secara logis struktur aplikasinya mengikuti prinsip 3-tier architecture, yaitu presentation Layer (Client) berupa aplikasi Next.js yang berperan sebagai antarmuka pengguna, business logic layer (Application Services) yang diwakili oleh layanan-layanan Supabase seperti autentikasi, aturan akses (RLS), dan manajemen API, dan data layer yang diwakili oleh database PostgreSQL dan Supabase Storage yang menyimpan seluruh data aplikasi. Dengan pendekatan ini, aplikasi menjadi ringan, efisien, mudah dikembangkan, dan cepat di-deploy tanpa harus mengelola infrastruktur backend tambahan. Supabase bertindak sebagai solusi backend lengkap (Backend-as-a-Service), menggabungkan logika bisnis dan manajemen data dalam satu platform cloud-native. Struktur arsitektur aplikasinya secara garis besar dapat digambarkan sebagai berikut.

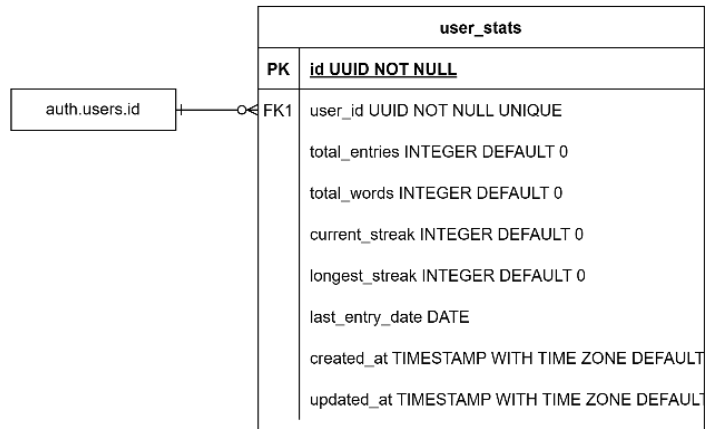
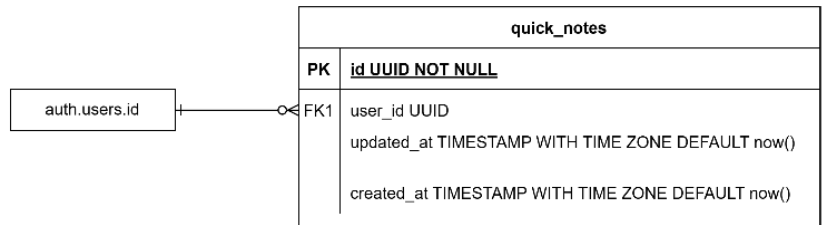
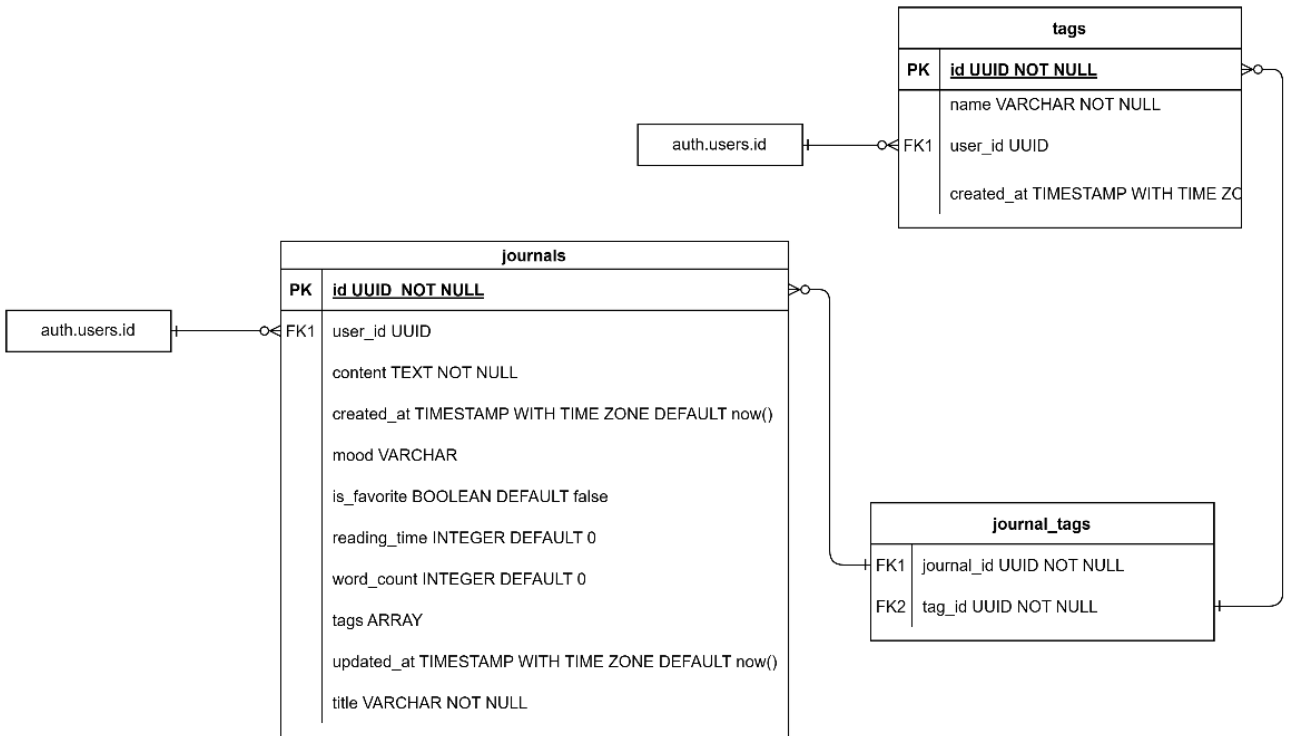


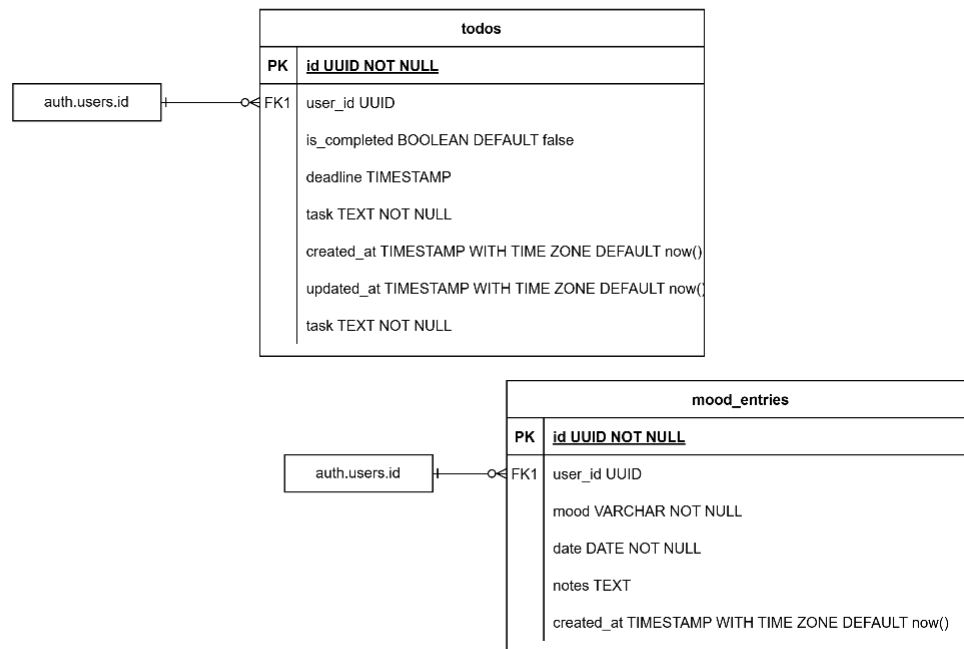
Gambar 1 Arsitektur Aplikasi

3.2. Pemodelan Data Menggunakan Entity-Relationship Diagram

Struktur basis data dirancang untuk mendukung berbagai fitur aplikasi, termasuk sistem jurnal harian, tagging, mood tracking, quick notes, todo list, dan statistik pengguna. Relasi antara

tabel mencerminkan kebutuhan akan fleksibilitas pencatatan dan pengorganisasian data oleh masing-masing pengguna. Berikut adalah representasi ERD yang disederhanakan.





Gambar 2 ER Diagram

3.3. Data Dictionary

Bagian ini menjelaskan struktur data (melalui tabel dan kolom database) serta menggambarkan perilaku sistem menggunakan diagram perilaku. Diagram perilaku bertujuan untuk menunjukkan bagaimana sistem merespons interaksi pengguna dan bagaimana alur proses terjadi dari sudut pandang pengguna atau sistem.

3.3.1. Struktur Data

Pada aplikasi journaling ini, terdapat beberapa tabel utama yang mendukung fitur-fitur kunci seperti pencatatan jurnal harian, pengelolaan tag, mood tracking, todo list, dan quick notes. Setiap tabel dirancang untuk menyimpan informasi spesifik, serta memiliki relasi antar tabel untuk menjaga integritas data dan mempermudah proses pengambilan informasi. Berikut adalah penjabaran setiap tabel beserta kolom-kolomnya, tipe data yang digunakan, dan fungsinya dalam sistem.

Tabel 2 Stuktur Data

Tabel	Kolom	Tipe Data	Deskripsi
journals	id	UUID (PK)	ID unik untuk setiap catatan jurnal
	title	VARCHAR	Judul jurnal

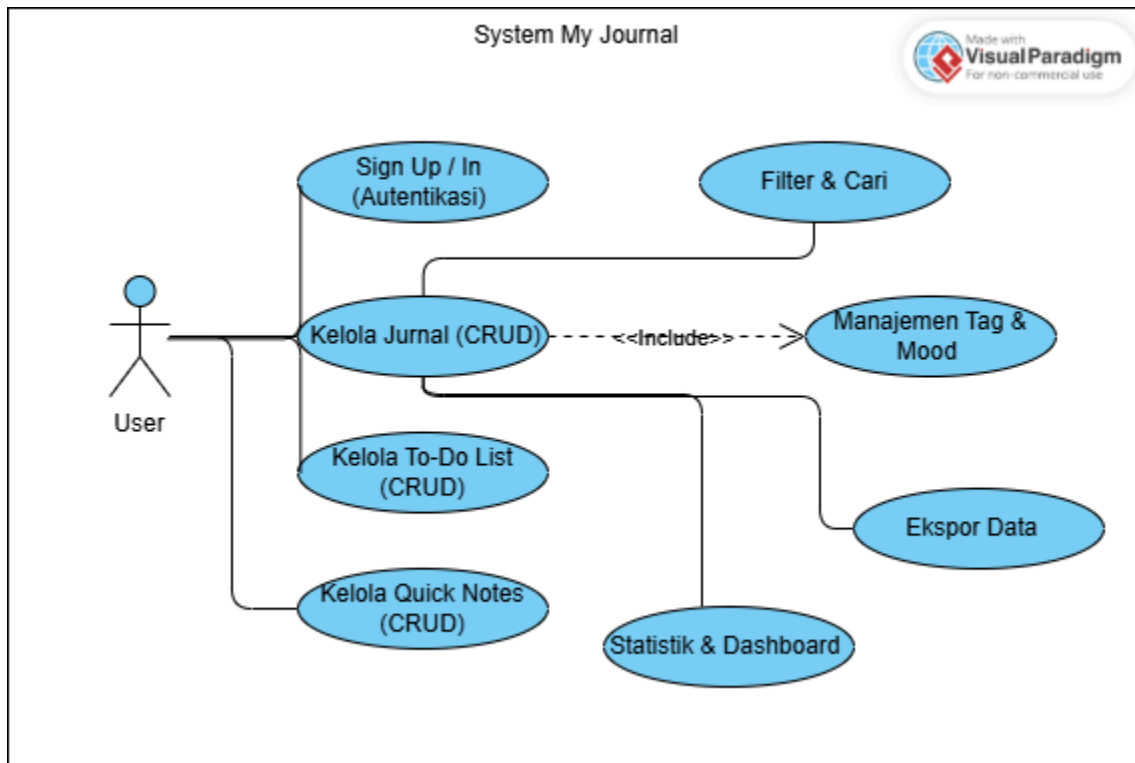
	content	TEXT	Isi jurnal lengkap
	mood	VARCHAR	Emoji atau teks mood
	is_favorite	BOOLEAN	Penanda favorit
	reading_time	INTEGER	Estimasi waktu baca (detik)
	word_count	INTEGER	Jumlah kata dalam jurnal
	user_id	UUID (FK)	ID pengguna yang memiliki jurnal
	created_at	TIMESTAMP	Tanggal dibuat
	updated_at	TIMESTAMP	Tanggal diubah
tags	id	UUID (PK)	ID tag unik
	name	VARCHAR	Nama tag
	user_id	UUID (FK)	ID pengguna yang membuat tag
	created_at	TIMESTAMP	Tanggal dibuat
journal_tags	journal_id	UUID (FK)	Relasi ke jurnal
	tag_id	UUID (FK)	Relasi ke tag
todos	id	UUID (PK)	ID tugas unik
	task	TEXT	Deskripsi tugas
	is_completed	BOOLEAN	Status selesai
	deadline	TIMESTAMP	Deadline tugas
	user_id	UUID (FK)	ID pengguna
	created_at	TIMESTAMP	Tanggal dibuat

	updated_at	TIMESTAMP	Tanggal diubah
quick_notes	id	UUID (PK)	ID note unik
	content	TEXT	Isi catatan cepat
	user_id	UUID (FK)	ID pengguna
	created_at	TIMESTAMP	Tanggal dibuat
	updated_at	TIMESTAMP	Tanggal diubah
mood_entries	id	UUID (PK)	ID mood unik
	mood	VARCHAR	Mood pengguna
	date	DATE	Tanggal mood
	notes	TEXT	Catatan tambahan
	user_id	UUID (FK)	ID pengguna
	created_at	TIMESTAMP	Tanggal dibuat
user_stats	id	UUID (PK)	ID statistik unik
	user_id	UUID (FK)	ID pengguna
	total_entries	INTEGER	Jumlah jurnal
	total_words	INTEGER	Total kata dari semua jurnal
	current_streak	INTEGER	Streak hari ini
	longest_streak	INTEGER	Rekor streak
	last_entry_date	DATE	Tanggal terakhir membuat jurnal
	created_at	TIMESTAMP	Tanggal dibuat

	updated_at	TIMESTAMP	Tanggal diubah
--	------------	-----------	----------------

3.3.2. Use Case Diagram

Untuk memahami interaksi antara pengguna dengan sistem secara menyeluruh, digunakan Use Case Diagram sebagai alat bantu visualisasi. Diagram ini menggambarkan hubungan antara aktor (pengguna) dengan berbagai fitur atau fungsionalitas utama dalam aplikasi digital journaling yang telah dikembangkan. Setiap use case merepresentasikan tindakan atau aktivitas yang dapat dilakukan oleh pengguna, mulai dari proses autentikasi, pengelolaan jurnal dan tugas harian, hingga fitur pendukung seperti filter, statistik, dan ekspor data. Dengan diagram ini, alur penggunaan sistem menjadi lebih jelas dan terstruktur, sehingga memudahkan dalam proses analisis maupun pengembangan lebih lanjut.



Gambar 3 Use Case Diagram

3.4. Perancangan User Interface

Antarmuka pengguna (User Interface / GUI) pada aplikasi ini merupakan sebuah web-based graphical interface yang dibangun menggunakan teknologi React (Next.js) dan Tailwind CSS yang dirancang dengan prinsip modern, minimalis, dan interaktif, memanfaatkan gaya visual glassmorphism untuk memberikan tampilan yang bersih, elegan, dan futuristik.

Desain UI sepenuhnya dibangun menggunakan Tailwind CSS, yang memungkinkan penggunaan utility-first classes untuk menghasilkan layout yang responsif dan mudah dikustomisasi.

Tampilan antarmuka dioptimalkan untuk penggunaan pada perangkat desktop dan laptop dengan spesifikasi rendah, sehingga ringan, cepat dimuat, dan tetap fungsional. Komponen-komponen utama seperti tombol aksi, form input, modal, dialog konfirmasi, dan kartu informasi dibuat dalam bentuk komponen reusable agar mudah digunakan kembali dan dipelihara dalam jangka panjang. Berikut beberapa elemen utama dalam desain UI.

- Layout dashboard yang menampilkan statistik pengguna secara real-time (jumlah entri, favorit, total kata, dll.).
- Navigasi sidebar untuk akses cepat ke halaman jurnal, todo list, quick notes, dan fitur ekspor.
- Komponen pencarian dan filter yang memungkinkan pengguna menelusuri entri berdasarkan tag, mood, tanggal, dan favorit.
- Form CRUD untuk jurnal, todo list, quick notes, dan mood entries, dirancang sederhana dengan validasi input yang jelas.
- Feedback interaktif seperti notifikasi sukses, error, dan loading spinner, untuk memberikan kejelasan pada tindakan pengguna.

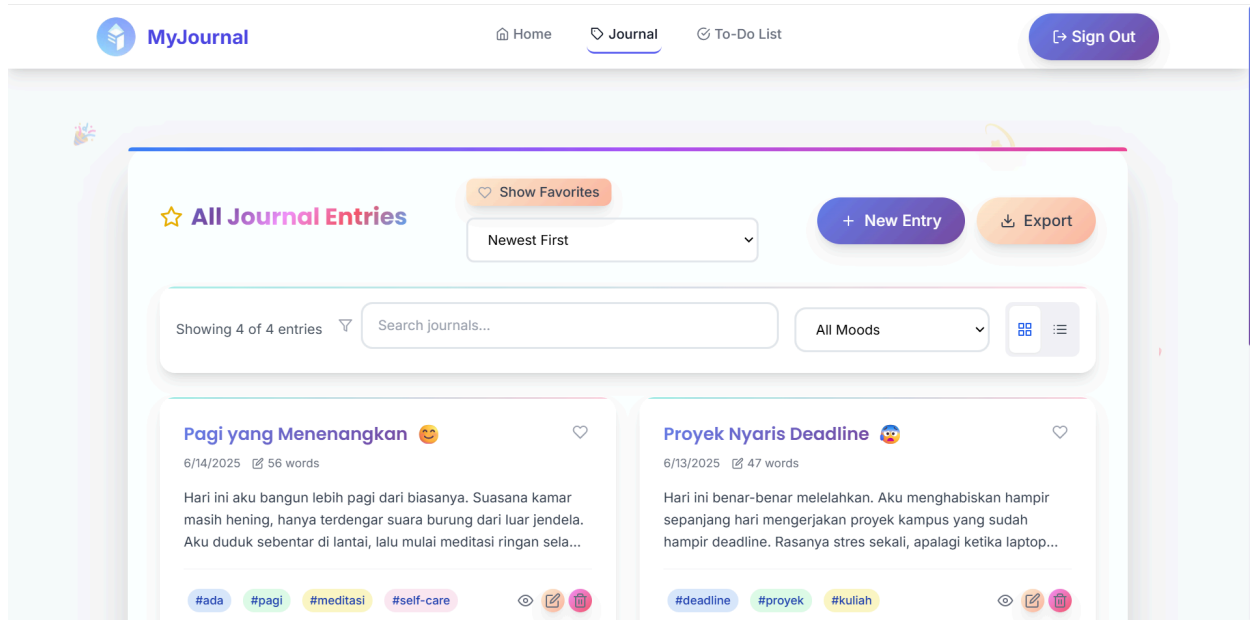
4. IMPLEMENTASI APLIKASI

Proses implementasi difokuskan pada penyelesaian modul-modul utama sesuai kebutuhan fungsional, mulai dari fitur pencatatan jurnal, pengelolaan tugas (to-do list), hingga ekspor data. Setiap modul didesain dengan mempertimbangkan aspek pengalaman pengguna, kecepatan, dan fleksibilitas penggunaan. Berikut adalah uraian komponen dan fitur yang telah berhasil diimplementasikan.

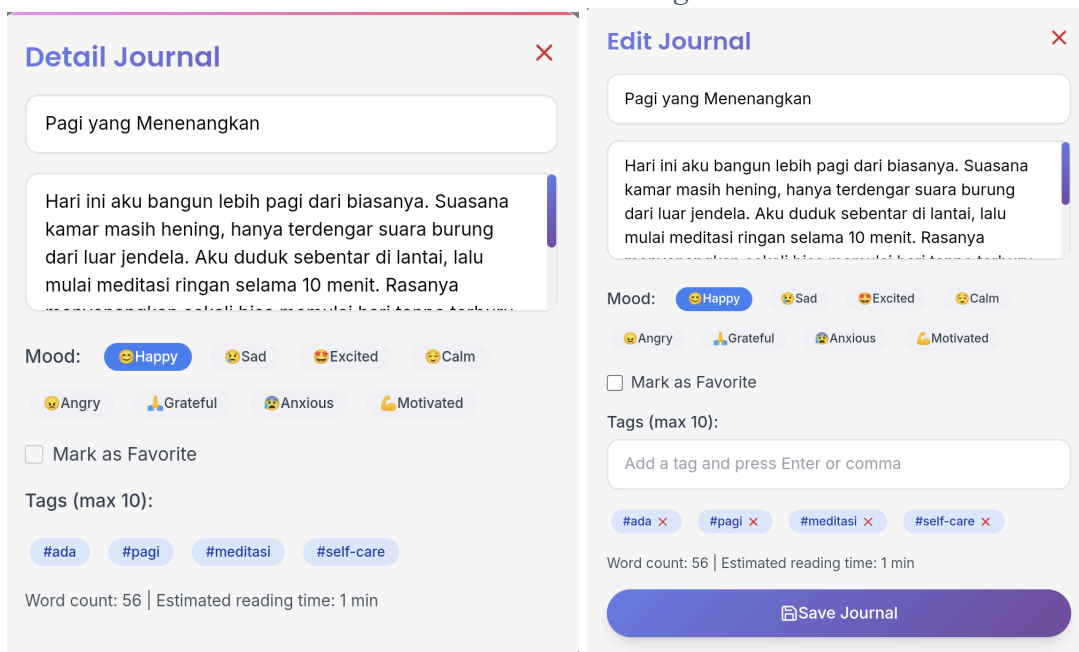
4.1. Modul Journaling

Modul journaling merupakan komponen utama dalam aplikasi ini yang memungkinkan pengguna untuk mencatat pengalaman harian mereka secara personal dan aman. Fitur-fitur utama yang telah berhasil diimplementasikan antara lain pembuatan, pembacaan, pengeditan, dan penghapusan (CRUD) entri jurnal. Setiap entri dapat dilengkapi dengan tag khusus untuk mengelompokkan isi catatan, serta ditandai dengan pilihan mood berbasis emoji yang memudahkan pengguna dalam merefleksikan suasana hati mereka saat menulis. Selain itu, terdapat fitur untuk menandai entri sebagai favorit agar lebih mudah diakses kembali. Pengguna

juga dapat memanfaatkan sistem filter yang fleksibel untuk menampilkan entri berdasarkan tag, tanggal, suasana hati, dan status favorit. Modul ini dibangun menggunakan Next.js di sisi frontend, dengan dukungan Tailwind CSS untuk styling antarmuka, dan Supabase sebagai backend untuk pengelolaan data. Semua data jurnal disimpan secara aman di dalam database PostgreSQL yang dikelola oleh Supabase, serta dilindungi menggunakan mekanisme autentikasi dan Row-Level Security.



Gambar 4 Journal Page



Gambar 5a Read dan Edit Journal Form

New Journal ✕

Title

Write your journal here...

Mood: 😊 Happy 😞 Sad 😄 Excited 😌 Calm
😡 Angry 🙏 Grateful 😟 Anxious 💪 Motivated

☐ Mark as Favorite

Tags (max 10):

Add a tag and press Enter or comma

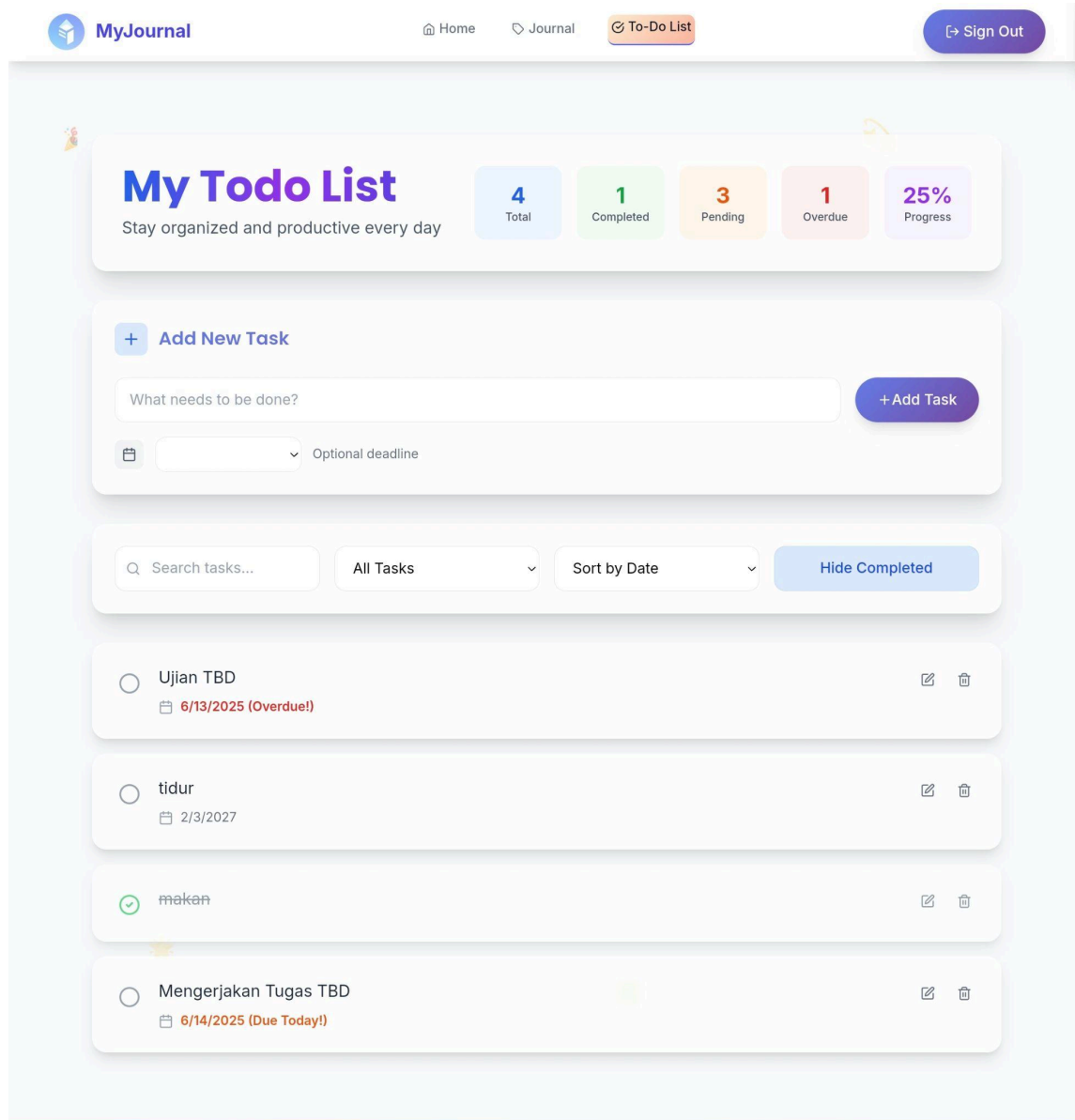
Word count: 0 | Estimated reading time: 0 min

Save Journal

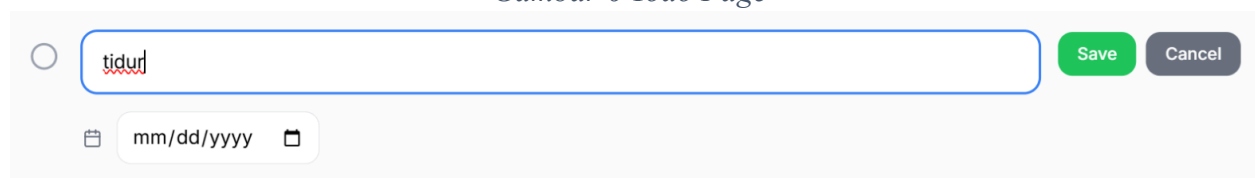
Gambar 5b Insert New Journal Form

4.2. Modul To-Do List

Modul to-do list berfungsi sebagai fitur tambahan untuk mendukung produktivitas pengguna. Fitur ini memungkinkan pengguna mencatat tugas-tugas yang harus diselesaikan, menyusun prioritas, dan melacak kemajuan harian mereka. Pengguna dapat menambahkan tugas baru, menandai tugas sebagai selesai, serta melakukan pengeditan atau penghapusan terhadap tugas yang telah ada. Tugas juga dapat diurutkan dan diklasifikasikan berdasarkan status penyelesaian. Selain itu, aplikasi menampilkan indikator progres visual yang memberikan gambaran umum mengenai jumlah tugas yang telah diselesaikan. Desain modul ini dibuat responsif agar tetap nyaman digunakan di perangkat seluler maupun desktop. Modul ini dibangun menggunakan React Hooks seperti `useState` dan `useEffect` untuk pengelolaan state secara lokal, dengan integrasi Supabase untuk penyimpanan data secara permanen.



Gambar 6 Todo Page

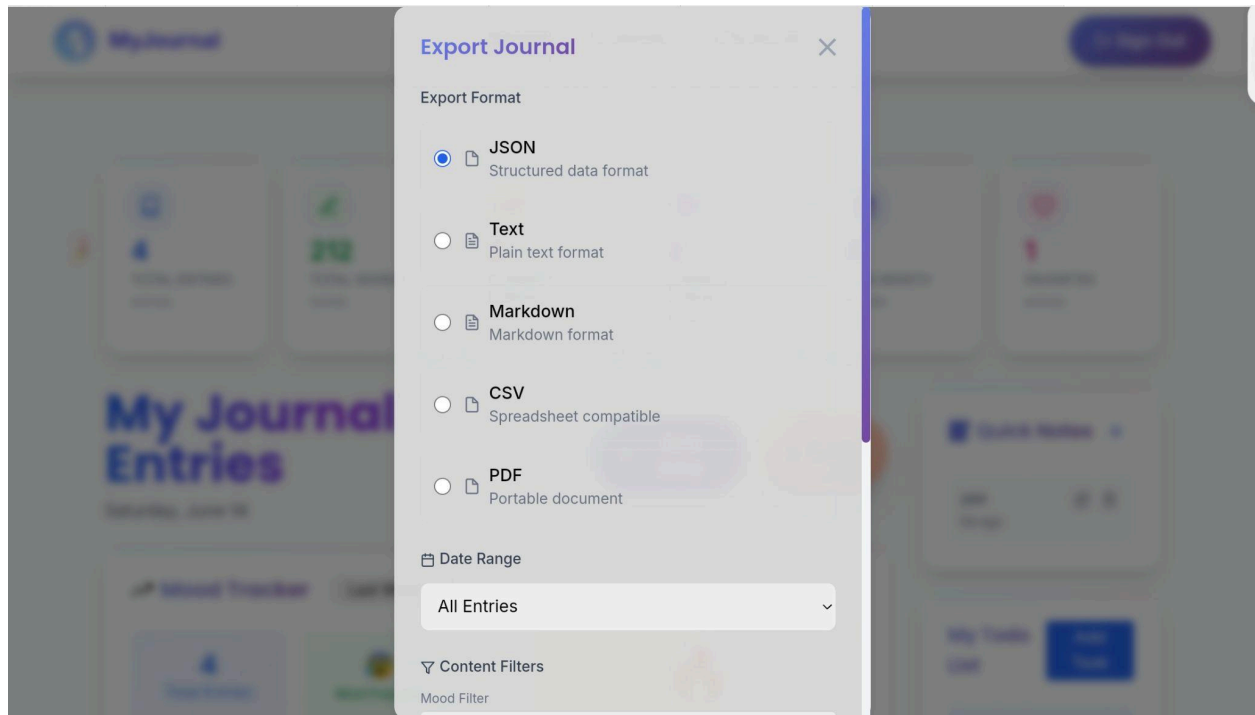


Gambar 7 Edit Todolist Form

4.3. Modul Export Data

Modul export data memberikan kemampuan bagi pengguna untuk melakukan backup atau membagikan data jurnal mereka dalam berbagai format file. Fitur ini memungkinkan pengguna untuk mengekspor entri jurnal lengkap beserta tag, mood, dan statistik terkait ke dalam

format seperti JSON, CSV, TXT, Markdown, dan PDF. Pengguna dapat memilih jenis data yang ingin disertakan dalam ekspor melalui dialog yang interaktif dan mudah digunakan. Fitur ini diimplementasikan dengan menggunakan JavaScript Blob API untuk menghasilkan file secara langsung di sisi klien, dan memanfaatkan query dari Supabase untuk mengambil data jurnal berdasarkan akun pengguna yang sedang aktif. Modul ini menjadi salah satu fitur penting untuk mendukung portabilitas data serta transparansi pengguna dalam mengelola catatan pribadinya.

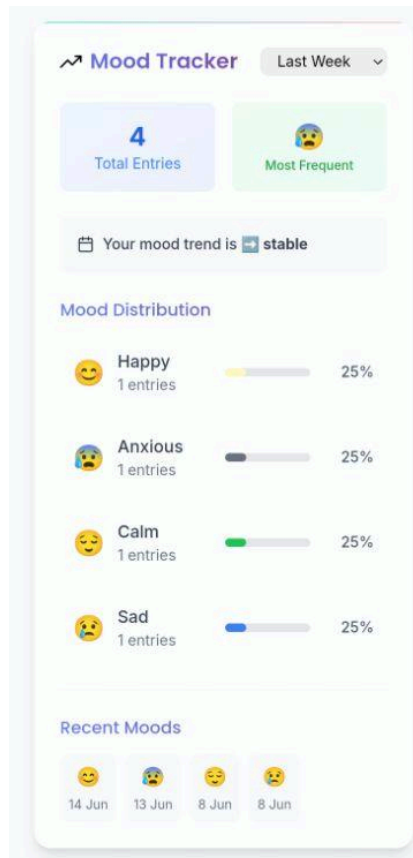


Gambar 8 Export Journal Form

4.4. Modul Mood Tracker

Modul mood tracker dirancang untuk membantu pengguna merefleksikan dan memantau suasana hati mereka dari waktu ke waktu. Setiap entri jurnal yang ditulis dapat dikaitkan dengan satu dari beberapa opsi mood yang telah disediakan, yang ditampilkan dalam bentuk emoji untuk memberikan pengalaman visual yang menyenangkan dan intuitif. Seluruh data mood yang dikumpulkan dari entri pengguna kemudian direkapitulasi dan divisualisasikan dalam grafik sederhana yang menggambarkan tren emosional pengguna sepanjang hari, minggu, atau bulan tertentu. Dengan fitur ini, pengguna dapat memperoleh pemahaman lebih baik mengenai pola suasana hati mereka dan bagaimana perasaan tersebut berkaitan dengan aktivitas atau catatan harian mereka. Mood tracker terintegrasi sepenuhnya dengan Supabase untuk penyimpanan data

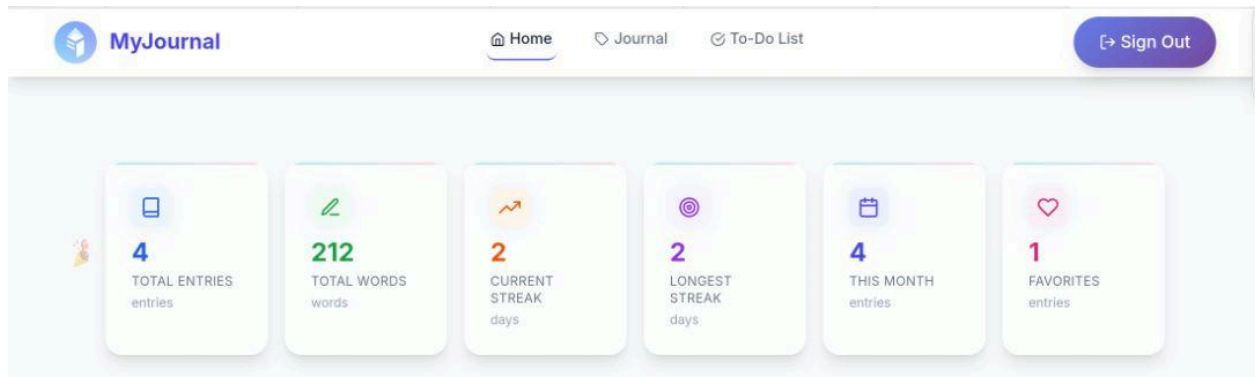
mood dan Next.js di sisi frontend, serta menggunakan pustaka visualisasi data ringan berbasis JavaScript untuk menampilkan grafik tren.



Gambar 9 Mood Tracker

4.5. Modul Dashboard Statistik

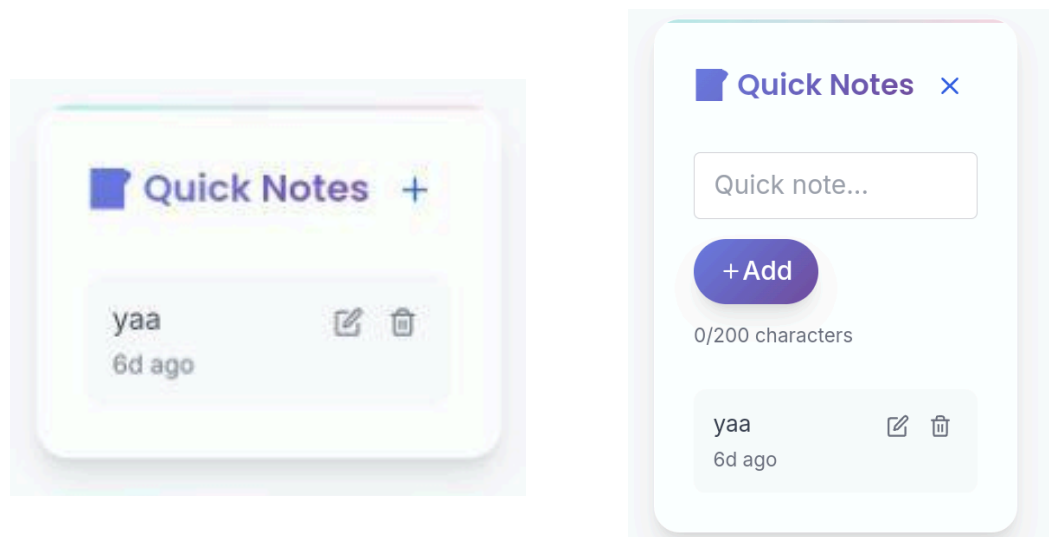
Dashboard statistik berfungsi sebagai pusat informasi yang merangkum aktivitas pengguna dalam aplikasi secara real-time. Modul ini menampilkan berbagai metrik penting seperti jumlah total entri jurnal, total kata yang telah ditulis, jumlah tag yang digunakan, entri favorit, serta grafik perkembangan streak atau konsistensi menulis harian. Fitur ini mendorong pengguna untuk lebih aktif dan konsisten dalam melakukan journaling dengan memberikan insight kuantitatif atas kebiasaan mereka. Statistik diperbarui secara otomatis seiring perubahan data di dalam database, dan disusun dalam bentuk kartu statistik serta grafik yang mudah dibaca. Modul ini memanfaatkan query langsung ke Supabase dengan optimisasi agar proses pemuatan data tetap cepat dan ringan. Komponen visual dibangun dengan pendekatan reusable untuk memudahkan pengembangan berkelanjutan.



Gambar 10 Dashboard Statistik

4.6. Modul Quick Notes

Modul quick notes merupakan fitur tambahan yang memungkinkan pengguna menulis catatan singkat secara cepat tanpa harus membuka form entri jurnal penuh. Fitur ini berguna untuk mencatat ide, pengingat, atau hal-hal penting yang bersifat sementara. Setiap catatan dapat disimpan, diedit, dan dihapus kapan saja, serta disimpan secara terpisah dari entri jurnal utama. Modul ini dibuat dengan tampilan sederhana namun tetap konsisten secara desain dengan bagian aplikasi lainnya, menjaga kenyamanan penggunaan. Teknologi yang digunakan mencakup komponen React berbasis useState dan useEffect untuk pengelolaan interaksi cepat, serta Supabase untuk menyimpan data note secara persistent per pengguna.

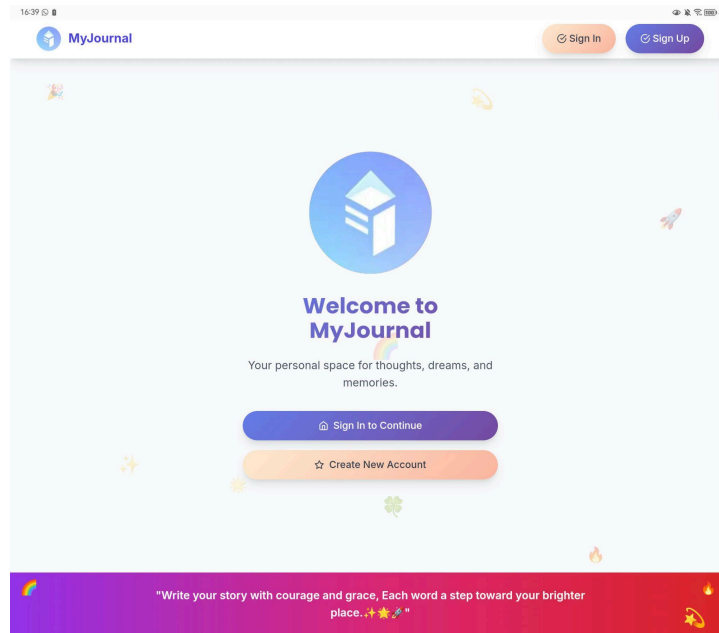


Gambar 11 Quick Notes Preview and Quick Notes Insert Form

4.7. Modul Autentikasi (Sign In & Sign Up)

Modul autentikasi pengguna dibangun menggunakan layanan Supabase Auth, yang menyediakan proses sign up dan sign in menggunakan email dan password. Setiap pengguna

hanya dapat mengakses data jurnal, tugas, dan catatan miliknya sendiri yang terhubung melalui User ID. Setelah login berhasil, seluruh data pengguna langsung dimuat berdasarkan session yang aktif. Fitur logout juga tersedia untuk mengakhiri sesi secara aman. Modul ini merupakan dasar dari sistem personalisasi data dan keamanan aplikasi secara keseluruhan.



Gambar 12 HomePage

LAMPIRAN

Lampiran 1. Struktur Proyek Aplikasi

```
|
+---src
    +---components
        DashboardStats.js
        EnhancedJournalModal.js
        ExportButton.js
        JournalStreaks.js
        Layout.js
        LoadingSpinner.js
        MoodTracker.js
        QuickNotes.js
        SearchFilter.js
        TodoProgressAndList.js
    +---context
        AuthContext.js
    +---pages
        favicon.ico
        index.js
        login.js
        signup.js
        tags.js
        todos.js
        _app.js
        _document.js
    +---styles
        globals.css
    +---utils
        supabaseClient.js
```

Lapiran 2. Source Code

Lapiran 2.1. Autentikasi Supabase

```

1 import { createContext, useContext, useEffect, useState } from 'react'
2 import { supabase } from '../utils/supabaseClient'
3
4 const AuthContext = createContext()
5
6 export function AuthProvider({ children }) {
7   const [user, setUser] = useState(null)
8   const [loading, setLoading] = useState(true)
9
10  useEffect(() => {
11    const getSession = async () => {
12      const {
13        data: { session },
14        error,
15      } = await supabase.auth.getSession()
16      if (error) {
17        console.error('Error fetching session:', error)
18      }
19      setUser(session?.user ?? null)
20      setLoading(false)
21    }
22
23    getSession()
24
25    const { data: authListener } = supabase.auth.onAuthStateChange((_event, session) => {
26      setUser(session?.user ?? null)
27      setLoading(false)
28    })
29
30    return () => {
31      authListener.subscription.unsubscribe()
32    }
33  }, [])
34
35  const signIn = async (email, password) => {
36    const { data, error } = await supabase.auth.signInWithPassword({ email, password })
37    if (error) throw error
38    return data.user
39  }
40
41  const signUp = async (email, password) => {
42    const { data, error } = await supabase.auth.signUp({ email, password })
43    if (error) throw error
44    return data.user
45  }
46
47  const signOut = async () => {
48    const { error } = await supabase.auth.signOut()
49    if (error) throw error
50  }
51
52  const value = {
53    user,
54    signIn,
55    signUp,
56    signOut,
57  }
58
59  return <AuthContext.Provider value={value}>{!loading && children}</AuthContext.Provider>
60 }
61
62 export function useAuth() {
63   return useContext(AuthContext)
64 }
65

```

Lapiran 2.1. Fungsi CRUD Jurnal


```

1  const handleSubmit = async (e) => {
2    e.preventDefault()
3    if (!title.trim() || !content.trim()) {
4      alert('Please fill in both title and content')
5      return
6    }
7
8    try {
9      setLoading(true)
10
11      const journalData = {
12        title: title.trim(),
13        content: content.trim(),
14        is_favorite: isFavorite,
15        word_count: wordCount,
16        reading_time: readingTime,
17        user_id: userId,
18        mood: mood
19      }
20
21      let journalId = journal?.id
22
23      if (journal) {
24        const { error } = await supabase
25          .from('journals')
26          .update(journalData)
27          .eq('id', journalId)
28
29        if (error) throw error
30      } else {
31        const { data, error } = await supabase
32          .from('journals')
33          .insert([journalData])
34          .select()
35          .single()
36
37        if (error) throw error
38        if (data) journalId = data.id
39      }
40
41      if (tags.length > 0) {
42        if (!journalId) throw new Error('journalId is undefined!')
43
44        const { error: delError } = await supabase
45          .from('journal_tags')
46          .delete()
47          .eq('journal_id', journalId)
48
49        if (delError) throw delError
50
51        for (const tagObj of tags) {
52          const tagName = tagObj.name
53          const { data: existingTags, error: tagError } = await supabase
54            .from('tags')
55            .select('id')
56            .eq('name', tagName)
57            .eq('user_id', userId)
58            .limit(1)
59
60          if (tagError) throw tagError
61

```

```

62     let tagId
63     if (!existingTags || existingTags.length === 0) {
64
65         if (tagName.length > 50) {
66             console.warn(`Tag "${tagName}" exceeds 50 characters and will be skipped.
67         `)
68             continue
69         }
70         const { data: newTags, error: insertTagError } = await supabase
71             .from('tags')
72             .insert([ { name: tagName, user_id: userId } ])
73             .select()
74             .single()
75
76         if (insertTagError) throw insertTagError
77
78         tagId = newTags.id
79     } else {
80         tagId = existingTags[0].id
81     }
82
83     const { error: journalTagError } = await supabase
84         .from('journal_tags')
85         .insert([ { journal_id: journalId, tag_id: tagId } ])
86
87     if (journalTagError) throw journalTagError
88 } else {
89     if (journalId) {
90         await supabase
91             .from('journal_tags')
92             .delete()
93             .eq('journal_id', journalId)
94     }
95 }
96
97 onClose(true)
98 resetForm()
99 } catch (error) {
100     console.error('Error saving journal:', error)
101     alert(`Failed to save journal entry: ${error.message}`)
102 } finally {
103     setLoading(false)
104 }
105 }

```

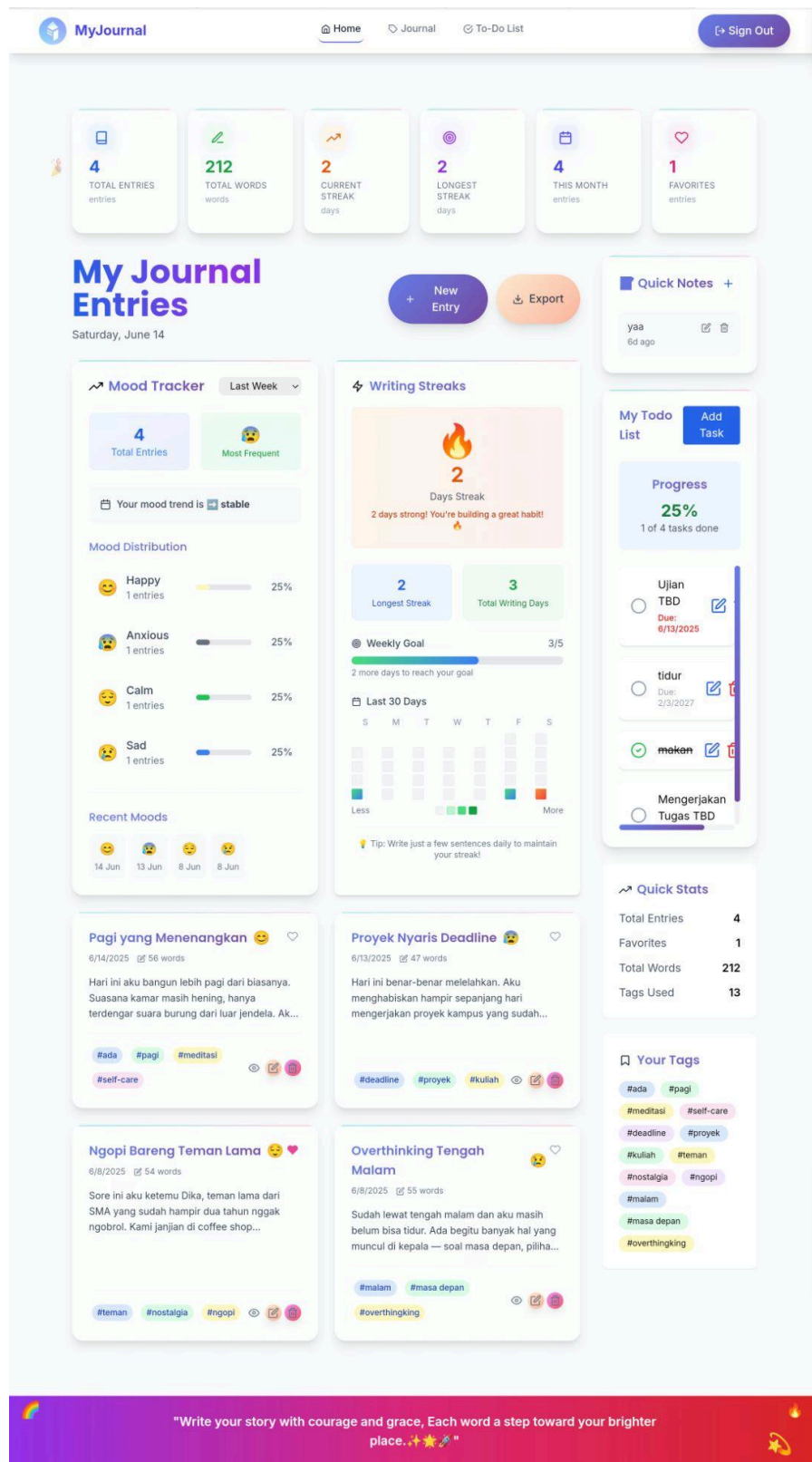
Lapiran 2.3. Integrasi dengan Supabase

```

1 import { createClient } from '@supabase/supabase-js'
2
3 const supabaseUrl = process.env.NEXT_PUBLIC_SUPABASE_URL
4 const supabaseAnonKey = process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY
5 console.log('Supabase URL:', process.env.NEXT_PUBLIC_SUPABASE_URL)
6 console.log('Supabase Key:', process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY?.slice(0, 10) +
7     '...')
8
9 export const supabase = createClient(supabaseUrl, supabaseAnonKey)

```

Lapiran 3. GUI Full Dashboard



Lapiran 4. Link Penting

GitHub: <https://github.com/Fannysbth/jornal-app>

Demo Aplikasi: <https://jornal-app-zeta.vercel.app/>