



Folio

Tiered Dynamic Bookstore Catalog

Mini Project SBD Pagi – Group 5 :



Diandra P. Wicaksono
2406342360



Marshal Aufa Diliyana
2406346913



Reinathan Ezkhiel K.
2406397675



Arkaan Pasya Seplitara
2406408073



Andhika Fadhlan W.
2306267164

PERIPLUS

Periplus merupakan toko buku daring terbesar di Indonesia. Mengelola ratusan ribu SKU buku dengan atribut yang sangat bervariasi, dari buku modern digital sampai arsip buku lama.

Konten

Ratusan ribu SKU buku dengan atribut yang sangat **heterogen**

Buku baru: e-book, voucher akses, CD resource → banyak field

Buku lama: hanya judul, pengarang, tahun terbit → minim field

Pain points dengan RDBMS

Schema Migration Berisiko

Setiap atribut baru pada buku modern memaksa **ALTER TABLE** yang berdampak ke jutaan baris buku lama.

Null Padding Massive

Buku lama tidak memiliki e-book, voucher, dan CD. Namun tabel tetap mengalokasikan **kolom kosong**.

Query lambat

Jutaan nilai **NULL** membengkakkan ukuran tabel dan **memperlambat** performa query secara keseluruhan.

Tidak fleksibel untuk Tearing

Buku baru kaya **fitur** sedangkan buku lama **minim data** dipaksa hidup di **skema yang sama** dan rigid.

Database : MongoDB Atlas

Schema Flexibility: Tiap dokumen dalam satu collection dapat memiliki struktur field yang berbeda. Buku baru dan lama bisa dalam satu collection books tanpa NULL-padding atau migrasi skema.

Horizontal Scalability: Mendukung sharding secara native, memungkinkan distribusi beban read ke banyak node seiring bertambahnya jumlah pengguna.

Embedding & Referencing: Memungkinkan data yang sering diakses bersama (ulasan terbaru) untuk di-embed langsung dalam dokumen buku, mengurangi jumlah round-trip ke database secara drastis.

Materialized View Native Support: Pola pre-aggregated summary document lebih mudah diimplementasikan di MongoDB dibanding MATERIALIZED VIEW pada basis data relasional yang memerlukan refresh terjadwal dan tetap terikat skema yang kaku.

Paradigma : Document Store

Mengapa Document Store?

Masalah **Periplus** adalah data yang **heterogen** strukturnya. **Document store** secara paradigma dirancang untuk menyimpan data semi-structured tanpa skema seragam.

Berbeda dengan key-value yang terlalu flat, wide-column yang masih schema-bound per column family, dan graph yang fokus ke relasi antar entitas.

Folio

Folio merupakan sistem katalog buku yang menerapkan arsitektur Data Tiering berbasis MongoDB. Setiap dokumen buku memiliki field shelf yang bernilai "hot shelf" untuk akses cepat atau "cold shelf" untuk hemat alokasi.

Problem that solved

Melayani katalog buku baru yang banyak atribut secara cepat

Menyimpan arsip buku lama tanpa membuang memori untuk field yang tidak relevan

Motivation

1. Buku baru dan lama tidak boleh diperlakukan sama di level storage
2. Halaman katalog harus termuat instan
3. Sistem harus bisa mengenali sendiri buku mana yang sedang "naik" dan mempromosikannya secara otomatis

System Architecture

Frontend

(React 18 + Vite + Tailwind + React Router + Axios)



Rest API

Backend

(Express.js + Node.js + Mongoose ODM + JWT + bcryptjs + node-cron)

- /api/auth
- /api/books
- /api/search
- /api/users

- weightedRating (Bayesian + time-decay)
- shelfManager (pruning + evaluateShelf)
- MaterializeViews



MongoDB Wire Protocol

Database

(MongoDB + Cron tiap 30 menit untuk shelf evaluation)

- books
- users
- bookbyusers

- clicklogs
- embeddedComments
- embeddedRatings

- text(title,author)
- genre
- registeredAt
- weightedRating
- {status, shelf}

Setiap interaksi user dari klik buku hingga rating **memicu pipeline backend** yang menambah clickLog, mem-prune embed, dan memperbarui weighted rating.

Pipeline Click → Recompute

- 1 Click** POST `/api/books/:id/click` → catat ipHash
- 2 Embed** Push clickLogs + increment totalViews/uniqueViewers
- 3 Prune** pruneClickLogs head+tail sesuai kuota hot/cold
- 4 Compute** computeWeightedRating Bayesian + time-decay
- 5 Recommend** updateRecommendation → simpan myRecommendation

API Endpoint Highlight · 22 routes

POST	<code>/api/auth/register</code>	—
POST	<code>/api/auth/login</code>	—
GET	<code>/api/books/:id</code>	—
POST	<code>/api/books/:id/click</code>	User
POST	<code>/api/books/:id/comment</code>	User
POST	<code>/api/books/:id/rate</code>	User
POST	<code>/api/books/:id/checkout</code>	User
POST	<code>/api/books/add</code>	Admin
GET	<code>/api/search/homepage</code>	—
GET	<code>/api/search/view/:viewType</code>	—

Folio memadukan **tiga strategi inti Document Store** untuk menyeimbangkan kecepatan baca, batas ukuran dokumen 16 MB, dan kebebasan agregasi.

Embedding

Comments, ratings, click logs hidup di dalam dokumen Book.

Akses cepat

satu round-trip MongoDB mengambil buku + ulasan.

Pruning otomatis

10 hot / 5 cold; head+tail click logs.

Trade-off

harus menjaga dokumen tetap di bawah 16 MB.

Referencing

BookByUsers menyimpan library tiap user secara terpisah.

Pola

```
{ userId, books:[ { bookId, acquiredAt, lastReadAt } ] }
```

Alasan

menghindari unbounded growth pada dokumen user.

Manfaat

checkout cepat tanpa menyentuh data buku.

Materialized View

Aggregation pipeline memproduksi view siap-pakai untuk katalog.

7 perspektif

yearGenre · genreAlpha · genreRegistered ...

Tambahan

hotBooks · mostCommented · mostViewed.

Strategi

render katalog ms-level tanpa JOIN.

Satu collection **books** menampung dua tier, diferensiasi terjadi pada level field, bukan tabel terpisah.

HOT SHELF

Top-10 weightedRating · totalComments · uniqueViewers

Comments	10 entri terbaru
Ratings	10 entri terbaru
Click logs	200 recent + 50 oldest
Atribut opsional	lengkap (e-book, voucher, audio)
Render	ditampilkan di hero & section HOT

COLD SHELF

Mayoritas katalog · arsip & long-tail

Comments	5 entri terbaru
Ratings	5 entri terbaru
Click logs	100 recent + 50 oldest
Atribut opsional	minimal (judul, author, year)
Render	muncul lewat search/filter

Cron node-cron tiap 30 menit memanggil evaluateShelf() – promosi 10 top → HOT, sisanya kembali COLD.

Skor buku dihitung dengan **Bayesian average + time-decay**, lalu disandingkan dengan sentiment komentar & tujuh materialized view siap-tampil.

Bayesian + Time Decay

$$\text{bayesian} = (v / (v + m)) \cdot R + (m / (v + m)) \cdot C$$

$$\text{weighted} = \text{bayesian} \cdot (0.8 + 0.2 \cdot \text{timeFactor})$$

$$\text{timeFactor} = 1 / (1 + \text{ageDays}/30)$$

- R** rata-rata rating buku ini
- v** jumlah rating (effective + sentiment)
- C** global mean rating semua buku
- m** prior konstanta (10) – anti-cold-start

Sentiment Pipeline

POSITIVE

NEUTRAL

NEGATIVE

`analyzeSentiment()` mencocokkan kata kunci EN/ID; label di-embed pada komentar dan menaikkan / menurunkan `effectiveCommentCount` untuk weighted rating.

7 Materialized Views · Aggregation Pipeline

- yearGenre
- genreRegistered
- genreRating
- yearRating
- genreAlpha
- registeredAlpha
- alphaRating

Pembangunan Folio menyingkap **empat trade-off** antara kecepatan, ukuran dokumen, kerumitan agregasi, dan ergonomi developer.

Embedding vs Referencing

Embed mempercepat baca tapi membatasi pertumbuhan. Kami batasi comments/ratings ke 10/5 entri dan pakai head+tail untuk click logs.

Hasil: dokumen tetap < 16 MB, satu round-trip cukup untuk halaman detail.

Soft Delete via Slot Recycling

Proposal awal merancang collection summary_blobs; kode memilih MongoDB Aggregation Pipeline langsung dari koleksi books.

Hasil: 7 view (genre · alfabet · tahun · rating) tanpa storage tambahan.

Threshold-based vs Ranking-based Tiering

Promosi cold→hot tidak memakai threshold 500-click; cron memilih top-10 menurut weightedRating, totalComments, uniqueViewers.

Hasil: HOT shelf selalu berisi 10 buku paling relevan saat ini.

Aggregation vs Pre-computed Blobs

DELETE tidak menghapus dokumen — field dasar di-unset & status='empty'. addBook me-recycle slot kosong.

Hasil: ObjectId stabil, riwayat library/checkout tidak putus.



A

ANDHIKA FADHLAN WIJANARKO

NPM 2306267164

**Backend Development & Database
Architecture**

KELOMPOK 5 · SBD PAGI

Tanggung Jawab Utama

- **Schema Design**

Merancang Book.js: shelf (hot/cold), embedded comments/ratings/clickLogs, optionalAttributes dinamis.

- **Tiering Engine**

Utils shelfManager.js (pruneClickLogs, pruneEmbeds, evaluateShelf) + cron node-cron tiap 30 menit.

- **Rating Engine**

weightedRating.js: Bayesian average dengan global mean C, prior $m=10$, time-decay $0.8 + 0.2 \cdot \text{timeFactor}$.

- **Indexing**

Membuat indeks text(title,author), genre, registeredAt, weightedRating, {status, shelf}.

FILES

backend/models/Book.js · utils/shelfManager.js ·
utils/weightedRating.js · server.js

M

MARSHAL AUFA DILIYANA

NPM 2406346913

Backend Development & API Integration

KELOMPOK 5 · SBD PAGI

Tanggung Jawab Utama

- **Controllers**

bookController, searchController, authController, userController – total 22 endpoint REST.

- **Aggregation Pipelines**

materializeViews.js (7 view + hotBooks + mostCommented + mostViewed).

- **Auth Middleware**

JWT verify, role guard adminOnly, slot recycling pada DELETE buku.

- **Click & Rate Flow**

Mengintegrasikan pruneClickLogs, updateRecommendation, dan computeWeightedRating pada setiap interaksi.

FILES

backend/controllers/*.js · routes/*.js · utils/materializeViews.js



R

REINATHAN EZKHEL

NPM 2406397675

Data Engineering & Database Seeding

KELOMPOK 5 · SBD PAGI

Tanggung Jawab Utama

- **Dataset Seeding**

seed.js menyiapkan ratusan buku awal dengan distribusi genre & tahun representatif (1985–2026).

- **Benchmark Script**

compare_neon.py membandingkan query MongoDB vs Neon (PostgreSQL) sebagai pembuktian paradigma.

- **Admin Bootstrap**

seedAdmin.js: auto-create akun admin@gmail.com pada startup server.

- **Embedded Reviews**

Memproduksi data dummy untuk embeddedComments & ratings agar Bayesian rating langsung valid.

FILES

backend/seed.js · compare_neon.py · utils/seedAdmin.js

A

ARKAAN PASYA SEPLITARA

NPM 2406408073

Frontend Development & UI/UX

KELOMPOK 5 · SBD PAGI

Tanggung Jawab Utama

- **Core Pages**

HomePage (hero slider auto-rotating), SearchPage (filter + paginasi), BookDetailPage, AdminPage.

- **Komponen Bersama**

Navbar, Footer, BookCard, StarRating, IntroScreen — dipakai konsisten antar halaman.

- **Tema Visual**

Glass-panel, kapsul rounded, animasi mesh-pulse, badge HOT SHELF, palet hitam-putih bermartabat.

- **Performance**

Promise.all paralel untuk 4 endpoint homepage; caching ringan via state untuk filter.

FILES

frontend/src/pages/*.jsx · components/*.jsx

D

DIANDRA PRAMESTI WICAKSONO

NPM 2406342360

Frontend Development & System Testing

KELOMPOK 5 · SBD PAGI

Tanggung Jawab Utama

- **Auth Pages**

LoginPage & RegisterPage — live password checklist (10 char, A-Z, a-z, 0-9, special) sinkron dengan regex backend.

- **Informational Pages**

AboutPage & ContactPage selaras dengan tema utama.

- **Axios Interceptor**

api/axios.js: lampirkan JWT, auto-logout pada 401, base URL Vite env.

- **End-to-End Testing**

Verifikasi alur register → login → click → rate → comment → checkout pada lima skenario user.

FILES

frontend/src/pages/LoginPage.jsx · RegisterPage.jsx · api/axios.js

<https://github.com/mrshlaf/folio>