

REKAYASA PERANGKAT LUNAK 2

PROPOSAL PENAWARAN PENGEMBANGAN PERANGKAT

LUNAK: SMART SHOPPING LIST APP



Kelompok 1:

Hafizhan Noor Amril (50422639)

Regina Prisilia Turana (51422403)

Stephen Helenus Ruswanto Kaawoan (51422549)

UNIVERSITAS GUNADARMA

JAKARTA

2026

1. PENDAHULUAN

A. Latar Belakang

Aktivitas belanja merupakan bagian dari kehidupan sehari-hari yang membutuhkan perencanaan agar berjalan efektif dan efisien. Tanpa perencanaan yang baik, pengguna sering mengalami berbagai permasalahan seperti lupa membeli barang yang diperlukan, membeli barang yang sama berulang kali, atau kesulitan mengingat barang yang pernah dibeli sebelumnya. Hal ini umumnya terjadi karena daftar belanja masih dicatat secara manual atau tidak memiliki sistem pencatatan yang terstruktur dan mudah diperbarui.

Perkembangan teknologi web membuka peluang untuk menghadirkan solusi digital yang lebih praktis dalam mengelola aktivitas belanja. Aplikasi web catatan belanja memungkinkan pengguna menyimpan, memperbarui, dan memantau daftar belanja secara real-time. Dengan dukungan fitur seperti status dibeli atau belum, pengelompokan berdasarkan kategori, prioritas item, serta pencarian dan filter, pengguna dapat mengelola daftar belanja dengan lebih terorganisasi dan mengurangi risiko kesalahan saat berbelanja.

Berdasarkan permasalahan tersebut, diperlukan sebuah aplikasi web catatan belanja yang tidak hanya fungsional tetapi juga mudah digunakan dan memiliki antarmuka yang modern. Aplikasi ini diharapkan mampu membantu pengguna dalam mengelola kebutuhan belanja secara lebih sistematis melalui fitur CRUD, UI responsif, serta arsitektur *client-server* yang andal. Dengan penerapan teknologi seperti React, Express, Prisma, dan SQLite, aplikasi ini dapat menjadi solusi efektif untuk meningkatkan kenyamanan dan efisiensi dalam aktivitas belanja sehari-hari.

B. Tujuan Proyek

1. Mengembangkan aplikasi web catatan belanja yang memungkinkan pengguna mencatat dan mengelola daftar belanja secara digital melalui fitur CRUD (*Create, Read, Update, Delete*).

2. Mempermudah pengguna dalam mengatur kebutuhan belanja dengan menyediakan fitur status dibeli/belum, kategori, kuantitas, satuan, prioritas, serta catatan tambahan agar daftar belanja lebih terstruktur dan jelas.
3. Menyediakan antarmuka pengguna yang modern dan responsif, termasuk dukungan *dark mode*, sehingga aplikasi nyaman digunakan di berbagai perangkat.
4. Meningkatkan efisiensi dan akurasi aktivitas belanja melalui fitur pencarian dan filter, sehingga pengguna dapat dengan cepat menemukan dan mengelompokkan item yang dibutuhkan.
5. Menerapkan arsitektur web yang maintainable dan terstruktur, menggunakan REST API, Prisma ORM, dan database SQLite untuk memastikan performa, kemudahan pengembangan, serta pemeliharaan aplikasi di masa depan.

2. RUANG LINGKUP DAN TEKNIS FITUR

A. Fitur Utama (*Functional Requirements*)

1. Sistem CRUD (*Create, Read, Update, Delete*)

Sistem menyediakan fungsi lengkap untuk mengelola item belanja, mulai dari menambahkan item baru dengan informasi detail seperti nama barang, kuantitas, satuan, kategori, prioritas, dan catatan tambahan. Pengguna dapat melihat seluruh daftar item dalam tampilan yang terstruktur, melakukan perubahan data jika terjadi penyesuaian kebutuhan, serta menghapus item yang sudah tidak diperlukan. Seluruh proses CRUD dilakukan secara *real-time* melalui REST API sehingga data selalu konsisten dengan *database*.

2. Manajemen Status Item.

Aplikasi menyediakan fitur *toggle* untuk menandai status item apakah sudah dibeli atau belum. Status ini ditampilkan secara visual menggunakan indikator atau perubahan gaya tampilan (misalnya coretan atau warna berbeda), sehingga pengguna dapat dengan cepat membedakan item yang sudah terpenuhi dan yang masih harus dibeli. Fitur ini membantu mencegah pembelian ganda dan meningkatkan efisiensi saat berbelanja.

3. Sistem Prioritas

Setiap item belanja dapat diberikan tingkat prioritas, seperti rendah, sedang, atau tinggi. Prioritas ditampilkan dalam bentuk label atau *badge* dengan warna berbeda agar mudah dikenali. Dengan adanya sistem ini, pengguna dapat menentukan item mana yang harus dibeli terlebih dahulu, terutama saat waktu atau anggaran terbatas.

4. Navigasi Canggih (Pencarian dan Filter)

Aplikasi dilengkapi dengan fitur pencarian berbasis kata kunci untuk memudahkan pengguna menemukan item tertentu dengan cepat. Selain itu, tersedia filter berdasarkan kategori, status (dibeli/belum), dan prioritas, sehingga daftar belanja dapat disesuaikan dengan kebutuhan pengguna. Fitur ini sangat membantu ketika jumlah item dalam daftar semakin banyak.

5. UI/UX Modern dan Responsif

Antarmuka aplikasi dirancang dengan tampilan modern, bersih, dan mudah dipahami, serta responsif di berbagai ukuran layar, baik *desktop* maupun perangkat *mobile*. Dukungan *dark mode* memungkinkan pengguna memilih tema tampilan yang nyaman di lingkungan minim cahaya, sekaligus meningkatkan kenyamanan visual dan pengalaman pengguna secara keseluruhan.

B. Spesifikasi Teknologi (*Tech Stack*)

1. *Frontend*: React/Vite
2. *Backend*: Node.js/Express
3. *Database*: SQLite
4. ORM: Prisma

3. METODOLOGI DAN LINI WAKTU

A. Metode Pengembangan

Metode yang paling sesuai untuk proyek aplikasi catatan belanja ini adalah Scrum (Agile) karena mampu memberikan kerangka kerja yang terstruktur namun tetap fleksibel. Meskipun skala proyek tergolong kecil hingga menengah, fitur-fitur yang ditargetkan cukup spesifik dan saling bergantung, seperti CRUD item, integrasi Prisma dengan SQLite, filter pencarian, serta dukungan *dark mode*. Scrum memungkinkan pengembangan dilakukan secara iteratif dan inkremental, sehingga setiap bagian aplikasi dapat dikembangkan, diuji, dan dievaluasi lebih awal tanpa harus menunggu keseluruhan sistem selesai.

Keunggulan Scrum pada proyek ini terletak pada fokusnya terhadap kualitas dan kemampuan beradaptasi. Melalui konsep Definition of Done, setiap fitur, misalnya *toggle* status dibeli atau validasi input item, hanya dianggap selesai apabila telah memenuhi kriteria fungsional dan non-fungsional yang ditetapkan. Selain itu, apabila terjadi perubahan kebutuhan di tengah pengembangan, seperti penyesuaian desain UI atau penambahan opsi filter, Scrum memungkinkan perubahan tersebut diakomodasi tanpa mengganggu keseluruhan rencana proyek.

Penerapan Scrum dalam proyek ini dibagi ke dalam dua *sprint* utama. Sprint pertama berfokus pada pembangunan fondasi sistem, meliputi setup database SQLite dengan Prisma, pembuatan REST API CRUD, serta tampilan dasar daftar item beserta fungsi tambah dan hapus. Sprint kedua berfokus pada penyempurnaan pengalaman pengguna melalui fitur pencarian dan filter, sistem prioritas dengan *badge*, *dark mode*, serta penataan UI yang lebih modern dan responsif. Dengan dukungan upacara Scrum yang ringkas seperti *sprint planning*, *review*, dan *retrospective*, proses pengembangan diharapkan berjalan terkontrol, kolaboratif, dan menghasilkan aplikasi yang siap digunakan secara optimal.

B. Struktur Sprint (*Backlog*)

Tabel 3. 1. Minggu 1: Arsitektur dan *Core CRUD (Data Flow)*

Tanggal	Kode	Deskripsi Pekerjaan	Output Utama
7-8 Desember 2025	SETUP	Inisialisasi React, Express, Prisma, & SQLite.	Arsitektur Folder & Koneksi DB.
9-10 Desember 2025	BACKEND	Pembuatan API CRUD & Validasi Input.	Endpoint API Teruji (Postman).
11-12 Desember 2025	FRONTEND	Form Input & List Display Utama.	UI Dasar & Integrasi API.
13 Desember 2025	REVIEW	Pengujian Skenario 1-4 (CRUD & Validasi).	Sistem CRUD Stabil.

Tabel 3. 2. Fitur Lanjutan dan Finalisasi UI

Tanggal	Kode	Deskripsi Pekerjaan	Output Utama
14-15 Desember 2025	LOGIC	Fitur Toggle Status & Sistem Prioritas.	Fungsi "Dibeli" & Badge Warna.
16-17 Desember 2025	NAV	Search Bar & Multi-Filter (Kategori/Status).	Pencarian & Filter Berhasil.
18 Desember 2025	STYLING	Implementasi Dark Mode & Responsivitas.	UI Modern (<i>Dark/Light</i>).
19 Desember 2025	QA/DOC	Data Seeding (db:seed) & Penulisan User Guide.	Paket Demo & Dokumentasi.
20 Desember 2025	RELEASE	Final Bug Fixing & Serah Terima Proyek.	Aplikasi Siap Kirim.

C. RENCANA PENGUJIAN

Berikut adalah enam poin utama yang harus dipenuhi dalam pengujian.

1. Input valid: item muncul.
2. Input kosong: muncul pesan *error* (validasi).
3. *Edit/delete*: perubahan tersimpan di SQLite (cek via Prisma Studio).
4. *Toggle*: klik status "Belum" berubah jadi "Dibeli".
5. *Search/filter*: menampilkan item yang relevan saja.
6. *Dark mode*: transisi warna halus saat mode diaktifkan.

D. INSTALASI DAN AKTIVASI

1. Ekstrak folder proyek dan buka di VS Code.
2. Jalankan perintah instalasi dan migrasi: npm install dan npx prisma migrate dev.
3. Jalankan aplikasi:
 - a. *Frontend*: <http://localhost:5173>
 - b. API Health Check: <http://localhost:4000/api/health>

E. TROUBLESHOOTING CEPAT

1. Jika database kosong, gunakan perintah: npm run db:seed.
2. Jika terjadi konflik port, pastikan port 4000 dan 5173 tidak sedang digunakan aplikasi lain.