

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Kajian terhadap sejumlah penelitian terdahulu yang berkaitan dengan sistem monitoring stok, pengelolaan gudang berbasis aplikasi mobile, pemanfaatan QR Code atau barcode sebagai identitas barang, integrasi data melalui REST API, serta pengembangan aplikasi lintas platform menunjukkan adanya beragam pendekatan teknologi dan metode pengembangan untuk meningkatkan efisiensi pencatatan inventori. Berbagai penelitian tersebut menjadi dasar dalam menentukan arsitektur aplikasi, mekanisme pemindaian, sinkronisasi data, dan metode pengujian pada penelitian ini. Untuk menunjukkan posisi penelitian “Aplikasi Monitoring Stok Barang Berbasis Flutter Menggunakan Pemindaian QR Code pada Hery Motor”, perbandingan fokus penelitian, teknologi, metode, hasil evaluasi, dan kesenjangan penelitian dirangkum dalam Tabel 2.1.

Penulis & Tahun	Judul/ Fokus Penelitian	Teknologi & Alat	Metode / Teori Utama	Matriks Evaluasi	Research Gap
-----------------	----------------------------	------------------	----------------------	------------------	--------------

Putra dan Halim, 2023 [20]	Smart Warehouse untuk pencatatan barang masuk, barang keluar, pengecekan stok, dan konfirmasi barang melalui web dan perangkat mobile.	Android, website, QR Code, barcode, database, serta pemisahan hak akses admin dan operator.	Mobil e-D dan Black Box Testin g.	Enam skenario yang dilaporkan terdiri atas 3 pengujian login, 2 pengujian input barang masuk, dan 1 pengujian konfirmasi barang keluar. Seluruh skenario berstatus “Berhasil .	Belum menggunakan Flutter. Alur sistem belum mencakup produksi, validasi mutasi Gudang 1–Gudang 2, kasir, packing, dan laporan terintegrasi seperti pada Hery Motor.
Noguerra , 2023 [21]	Desain dan evaluasi aplikasi monitoring inventori berbasis mobile menggunakan QR Code.	Mobile application, QR Code, JavaScript, qrcode.js, ZXing, basis data, backend, dan integrasi API.	Perancangan sistem dan evaluasi pada enam aspek kualitas.	Akurasi 4/5; efisiensi 5/5; usability 4/5; efektivitas 5/5; keamanan 5/5; maintain ability 4/5. Nilai rata-rata keseluruhan 4,5/5.	Tidak menggunakan Flutter dan belum membahas alur produksi, validasi antargudang, kasir, serta packing pesanan.
Wahyudi et al., 2024 [22]	Sistem persediaan berbasis Android dengan	Android Studio, smartphone, barcode scanner	Waterfall dan Black Box	Delapan skenario pengujian—login valid,	Menggunakan barcode, bukan QR Code, dan dikembangkan

	pemindaian barcode untuk pengecekan dan pembaruan stok.	pada aplikasi, PHP, MySQL, Use Case Diagram, dan Class Diagram.	Testing.	login tidak valid, cek stok, pencarian, pemindaian barcode, tambah barang, ubah barang, dan pencarian tidak ditemukan—seluruhnya berstatus “Valid” (8/8).	n dengan Android Studio, bukan Flutter. Belum menjelaskan sinkronisasi melalui REST API dan mutasi antargudang.
Jošt dan Taneski, 2025 [23]	Perbandingan framework pengembangan aplikasi mobile lintas platform, terutama Flutter dan React Native.	Flutter, React Native, .NET MAUI, Stack Overflow Survey, JetBrains Survey, Google Trends, GitHub, data lowongan kerja, R 4.3.3, dan RStudio.	Studi komparatif; Mann–Whitney U Test dan Pearson Correlation.	Perbedaan popularitas Flutter–React Native tidak signifikan: $p=0,59$ (Stack Overflow) dan $p=0,396$ (JetBrains). Minat pengembangan juga tidak signifikan: $p=0,818$.	Tidak membangun sistem inventori dan tidak mengevaluasi QR Code, transaksi stok, sinkronisasi server, atau proses gudang.

				Kepuasan jangka panjang berbeda signifikan: $p=0,004$; median Flutter 68,1 dan React Native 57,2.	
Ananda et al., 2024 [24]	Sistem informasi inventory berbasis website untuk data barang masuk, barang keluar, dan laporan stok.	Website, PHP, MySQL, UML, dan basis data terpusat.	Waterfall dan Black Box Testing.	Lima skenario login/logout dan enam skenario pengelolaan data barang dilaporkan berstatus “Berhasil” (11/11 skenario).	Berbasis website serta tidak menggunakan Flutter, QR Code melalui kamera smartphone, atau integrasi aplikasi mobile melalui REST API.
Lukas et al., 2024 [25]	Manajemen stok bahan baku gudang berbasis web dengan penerapan REST API.	Aplikasi web, REST API, basis data, fungsi CRUD, pencarian, ekspor Excel, dan laporan barang masuk-keluar.	Extreme Programming dan Black Box Testing.	Artikel menyatakan seluruh fungsi Black Box berjalan dengan benar dan aplikasi layak digunakan, tetapi tidak menyajikan	Berbasis web dan menggunakan Extreme Programming. Belum menggunakan Flutter, QR Code, maupun validasi pergerakan barang antargudang.

				an nilai numerik waktu respons, akurasi, atau persentase keberhasilan.	
--	--	--	--	--	--

2.2 LANDASAN TEORI

2.2.1 Manajemen Inventaris dan Gudang

Manajemen inventaris merupakan kegiatan pengelolaan persediaan barang yang mencakup proses pencatatan, penyimpanan, pengawasan, serta pengendalian stok untuk memastikan ketersediaan barang sesuai dengan kebutuhan operasional perusahaan. Sistem inventaris yang baik berfungsi untuk menyediakan informasi mengenai jumlah dan kondisi barang secara akurat sehingga perusahaan dapat melakukan perencanaan serta pengendalian persediaan secara lebih efektif [26]. Penerapan sistem informasi inventaris berbasis digital dapat membantu proses pengelolaan gudang melalui pencatatan transaksi yang lebih terstruktur, penyimpanan data secara terpusat, serta kemudahan dalam melakukan pemantauan perubahan stok barang [27]. Selain itu, digitalisasi sistem inventaris dapat membantu menggantikan proses pencatatan konvensional dengan sistem terkomputerisasi sehingga pengelolaan data barang menjadi lebih efektif dan mudah

dilakukan [28]. Adapun gambar visual mengenai manajemen inventaris dan gudang dapat dilihat pada Gambar 2.1.



Gambar 2. 1Manajemen Inventaris dan Gudang

2.2.2 Framework Flutter

Flutter merupakan framework pengembangan aplikasi yang dikembangkan oleh Google untuk membangun aplikasi lintas platform menggunakan satu basis kode (*single codebase*) sehingga aplikasi dapat dikembangkan pada berbagai platform dengan struktur kode yang sama [29]. Flutter menggunakan bahasa pemrograman Dart sebagai bahasa utama dalam proses pengembangan aplikasi serta menerapkan konsep widget sebagai komponen utama dalam membangun antarmuka pengguna yang fleksibel dan responsif [30]. Selain itu, Flutter memiliki fitur *Hot Reload* yang memungkinkan perubahan kode dapat langsung ditampilkan pada aplikasi tanpa harus melakukan proses kompilasi ulang secara keseluruhan sehingga membantu mempercepat proses pengembangan aplikasi mobile [31]. Dalam penelitian ini, Flutter digunakan sebagai framework utama untuk membangun aplikasi

monitoring stok barang karena mendukung pengembangan aplikasi mobile serta dapat digunakan untuk membangun aplikasi yang terintegrasi dengan berbagai fitur perangkat [32]. Adapun gambar visual mengenai framework Flutter dapat dilihat pada Gambar 2.2.



Gambar 2. 2Framework Flutter

2.2.3 QR Code (Quick Response Code)

QR Code merupakan teknologi kode dua dimensi yang mampu menyimpan informasi dalam bentuk pola matriks dan dapat dibaca menggunakan perangkat kamera digital seperti smartphone. QR Code memiliki keunggulan dibandingkan barcode satu dimensi karena mampu menyimpan informasi dengan kapasitas lebih besar serta memiliki fitur koreksi kesalahan (error correction) yang memungkinkan kode tetap dapat terbaca meskipun sebagian area mengalami kerusakan [33]. Dalam sistem inventaris, QR Code digunakan sebagai identitas unik barang yang dapat mempercepat proses pencarian, identifikasi, dan pencatatan data melalui mekanisme pemindaian otomatis, sehingga membantu meningkatkan akurasi pencatatan serta mengurangi kesalahan akibat proses input manual [34]. Adapun gambar visual mengenai QR Code dapat dilihat pada Gambar 2.3.



Gambar 2. 3QR Code

2.2.4 MySQL

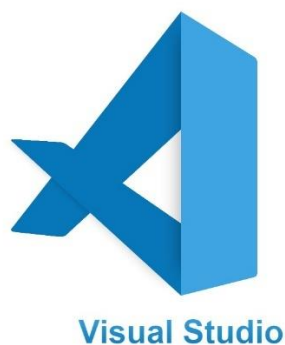
MySQL merupakan sistem manajemen basis data relasional (Relational Database Management System/RDBMS) yang digunakan untuk menyimpan, mengelola, dan mengolah data menggunakan bahasa SQL (Structured Query Language) [35]. Dalam aplikasi monitoring stok barang, database digunakan untuk menyimpan informasi pengguna, data barang, transaksi barang masuk, transaksi barang keluar, serta riwayat perubahan stok, sehingga penggunaan database yang terintegrasi membantu menjaga konsistensi data dan meningkatkan efisiensi operasional dalam pencatatan serta pemantauan barang secara akurat [36]. Adapun gambar visual mengenai MySQL dapat dilihat pada Gambar 2.3.



Gambar 2. 4MySQL

2.2.5 Visual Studio Code

Visual Studio Code (VS Code) merupakan perangkat lunak penyunting kode sumber (code editor) yang dikembangkan oleh Microsoft, bersifat sumber terbuka, dan dapat dijalankan di berbagai sistem operasi [37]. Perangkat lunak ini memiliki berbagai fitur unggulan seperti IntelliSense untuk melengkapi kode secara cerdas, fitur debugging, serta dukungan ekstensi yang luas untuk bahasa pemrograman Dart dan framework Flutter [38]. Peneliti menggunakan VS Code sebagai lingkungan pengembangan utama karena performanya yang ringan namun sangat bertenaga untuk mengelola struktur proyek, menulis logika program aplikasi monitoring, hingga proses integrasi database MySQL. Adapun gambar visual mengenai VS Code dapat dilihat pada Gambar 2.3.



Gambar 2. 5 Visual Studio Code

2.2.6 UML (Unified Modeling Language)

Unified Modeling Language (UML) merupakan sebuah bahasa pemodelan visual yang digunakan untuk menspesifikasikan,

memvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak. Pemodelan ini memberikan gambaran yang jelas mengenai sistem yang akan dibangun, baik dari sisi struktural maupun fungsional. Beberapa diagram UML yang digunakan dalam perancangan sistem ini meliputi:

1. *Activity Diagram*: Menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, keputusan (*decision*) yang mungkin terjadi, dan bagaimana alur tersebut berakhir. Notasi yang digunakan dalam perancangan *use case diagram* pada penelitian ini dapat dilihat pada Tabel 2.1.

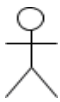
Table 2.1 Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Activity Final Node</i>	Bagaimana objek

NO	GAMBAR	NAMA	KETERANGAN
			dibentuk dan dihancurkan
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

2. Use Case Diagram: Merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem informasi yang akan dibuat antara aktor dan sistem. Notasi yang digunakan dalam perancangan activity diagram pada penelitian ini dapat dilihat pada Tabel 2.2.

Table 2.2 Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang

NO	GAMBAR	NAMA	KETERANGAN
			ada di atasnya objek induk (<i>ancestor</i>).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).

NO	GAMBAR	NAMA	KETERANGAN
10.		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

3. *Sequence Diagram*: Menggambarkan interaksi antar objek yang menunjukkan seorang aktor sedang melakukan pengelolaan data dalam satu skenario tertentu secara berurutan. Notasi-notasi tersebut disajikan pada Tabel 2.3.

Table 2.3 Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi.
2.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

4. *Class Diagram*: Diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang sedang digunakan, termasuk hubungan antar tabel pada basis data. Notasi-notasi tersebut disajikan pada Tabel 2.4.

Table 2.4 Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

NO	GAMBAR	NAMA	KETERANGAN
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya