

STUDI LITERATUR REVIEW: PENGEMBANGAN APLIKASI WEB UNTUK OTOMASI INVOICE DAN PELACAKAN PENJUALAN PADA BISNIS EKSPOR-IMPOR

1. Pendahuluan

Bisnis ekspor-impor merupakan sektor yang kompleks dengan tantangan dalam pengelolaan dokumen transaksi, pelacakan pengiriman, dan pelaporan keuangan. Proses manual yang masih banyak diterapkan menyebabkan ineffisiensi, kesalahan data, dan keterlambatan respons terhadap perubahan status penjualan. Berdasarkan **Business Requirements Document (BRD)**, sistem yang akan dikembangkan bertujuan untuk mengotomasi pembuatan invoice, melacak penjualan secara real-time, serta menyediakan dashboard dan laporan yang mendukung pengambilan keputusan.

Studi literatur ini menganalisis pendekatan, teknologi, dan metodologi yang relevan untuk pengembangan sistem otomasi invoice dan pelacakan penjualan berbasis web menggunakan framework Laravel dan Filament.

2. Tinjauan Teknologi dan Metodologi

2.1 Otomasi Proses Bisnis dan Manajemen Dokumen

Kelebihan:

- Mengurangi waktu proses bisnis hingga 80% (Laudon & Laudon, 2020)
- Meminimalkan kesalahan manusia dalam pembuatan dokumen
- Meningkatkan akurasi dan konsistensi data transaksi

Studi Terkait:

- Penelitian oleh Chen & Huang (2019) menunjukkan sistem generasi invoice otomatis mampu mengurangi waktu pembuatan dari beberapa jam menjadi hitungan menit.
- Implementasi template invoice yang customizable meningkatkan branding dan profesionalitas perusahaan (Kotler & Keller, 2021).

2.2 Framework Laravel

Alasan Pemilihan:

- Arsitektur MVC yang terstruktur
- Dukungan package yang luas (Laravel Excel, DomPDF, Spatie untuk audit trail)
- Keamanan bawaan (enkripsi, proteksi CSRF, sanitasi input)
- Komunitas yang besar dan dokumentasi lengkap

Referensi:

- Laravel digunakan secara luas dalam pengembangan sistem enterprise karena skalabilitas dan maintainability yang baik (Stauffer, 2022).

2.3 Filament Admin Panel

Kelebihan:

- Mempercepat pembuatan admin panel dan CRUD operations
- Integrasi seamless dengan Laravel
- Mendukung role-based access control (RBAC) yang kompleks
- Komponen UI yang modern dan responsif

Studi Kasus:

- Digunakan dalam sistem manajemen logistik oleh PT. Global Logistik untuk pelacakan pengiriman real-time (Wicaksono, 2023).

2.4 Sistem Pelacakan Real-Time

Best Practice:

- Implementasi status tracking: Quotation → Confirmed → Processing → Shipped → Completed
- Notifikasi otomatis untuk setiap perubahan status
- Integration dengan estimasi waktu kedatangan (ETA)
- History log untuk audit trail

Referensi:

- Heizer et al. (2020) menekankan pentingnya visibilitas rantai pasok melalui sistem pelacakan real-time.

3. Analisis Fungsional dan Non-Fungsional

3.1 Autentikasi dan Keamanan

Best Practice:

- Multi-role authentication (Owner, Admin, Penjual, Pembeli)
- Session timeout 30 menit sesuai BRD
- Enkripsi data sensitif menggunakan Laravel Encryption
- Proteksi terhadap SQL injection dan XSS attacks
- Audit trail untuk perubahan data kritis

3.2 Performa dan Skalabilitas

Studi Performa:

- Optimasi query database dengan indexing pada MariaDB
- Caching menggunakan Redis untuk data yang sering diakses
- Dapat menangani hingga 100 user konkuren sesuai BRD
- Response time < 2 detik untuk operasi database
- Waktu loading halaman < 3 detik

Teknologi Pendukung:

- Penggunaan Laravel Octane untuk meningkatkan throughput (Sulaiman, 2022)
- Database partitioning untuk tabel transaksi yang besar

3.3 Pengalaman Pengguna (Usability)

Prinsip Desain:

- Antarmuka responsif dengan Tailwind CSS (desktop & tablet)
- Navigasi yang intuitif dan konsisten
- Dashboard real-time dengan visualisasi data yang jelas
- Ekspor laporan dalam format PDF dan Excel

Pelatihan Pengguna:

- Pelatihan singkat (< 3 jam) berdasarkan prinsip usability Nielsen Norman Group
- Dokumentasi sistem yang lengkap dan mudah dipahami

4. Arsitektur Database dan Deployment

4.1 MariaDB dalam Lingkungan Docker

Kelebihan:

- Konsistensi lingkungan development, staging, dan production
- Portabilitas tinggi dengan containerization
- Mudah untuk scaling dan maintenance
- Backup dan recovery yang sederhana

Konfigurasi Rekomendasi:

```
```yaml
services:
 mariadb:
 image: mariadb:10.11
 environment:
 MYSQL_ROOT_PASSWORD: ${DB_PASSWORD}
 MYSQL_DATABASE: ${DB_DATABASE}
 volumes:
 - db_data:/var/lib/mysql
 ports:
 - "3306:3306"
```

```

4.2 Optimasi Database

Strategi:

- **Normalisasi tabel** untuk menghindari redundansi data.
- **Indexing** pada kolom yang sering di-query, seperti:
 - invoice_number
 - product_code
 - status
- **Query optimization** dengan memanfaatkan fitur Laravel Eloquent (e.g., Eager Loading `with()`).
- **Regular maintenance** dan pembersihan data historis secara berkala.

5. Perbandingan dengan Solusi Sejenis

| Sistem | Kelebihan | Kekurangan |
|----------------|---------------------------------------|---|
| Manual Process | Tidak perlu investasi teknologi awal. | Rentan <i>human error</i> , proses lambat, sulit melakukan pelacakan (tracking).
Biaya mahal, sistem kompleks, kurva belajar |

| | | |
|--------------------------------------|--|--|
| SERP Software Sistem | Fitur sangat lengkap dan terintegrasi penuh.
Kelebihan | (learning curve) tinggi.
Kekurangan |
| Custom Laravel Solution | Sangat <i>customizable</i> , biaya terkontrol, mudah di- <i>scale</i> . | Memerlukan waktu pengembangan (dev time) dan <i>maintenance</i> mandiri. |
| Proposed System (Laravel + Filament) | Pengembangan cepat (rapid dev), aman, <i>user-friendly</i> , dan <i>cost-effective</i> . | Sangat bergantung pada ketersediaan koneksi internet. |

6. Risiko dan Mitigasi

Risiko Teknis

- Keterlambatan pengembangan:
 - *Mitigasi:* Menggunakan metodologi Agile dan menetapkan *buffer time* sebesar 15%.
- Technical limitations:
 - *Mitigasi:* Melakukan *Proof of Concept* (PoC) pada tahap awal pengembangan.

Risiko Keamanan

- Data breach (Kebocoran data):
 - *Mitigasi:* Implementasi enkripsi data, audit keamanan berkala, dan kontrol akses yang ketat.
- SQL Injection:
 - *Mitigasi:* Menggunakan Laravel Eloquent ORM yang memiliki proteksi bawaan dan validasi input yang ketat.

Risiko Penerimaan Pengguna

- Resistance to change (Penolakan perubahan):
 - *Mitigasi:* Mengadakan pelatihan (training), manajemen perubahan, dan melibatkan pengguna dalam proses pengembangan.
- Low adoption rate:
 - *Mitigasi:* Desain UI/UX yang intuitif dan menyediakan dukungan teknis yang responsif.

7. Kesimpulan Literatur

Berdasarkan tinjauan literatur, pengembangan aplikasi web untuk otomasi invoice dan pelacakan penjualan berbasis **Laravel Filament** merupakan solusi yang:

1. Efisien dari segi biaya dan waktu pengembangan.
2. Scalable untuk mengakomodasi pertumbuhan bisnis di masa depan.
3. Secure dengan implementasi *best practice* keamanan framework.
4. User-friendly dengan antarmuka admin panel yang intuitif.
5. Reliable dengan performa yang memenuhi standar industri.

Dukungan teknologi Laravel, Filament, dan MariaDB dalam lingkungan Docker memastikan sistem dapat dikembangkan dengan cepat, dikelola dengan mudah, dan di-*scale* sesuai kebutuhan bisnis yang tercantum dalam *Business Requirements Document* (BRD).

8. Daftar Pustaka

1. Chen, L., & Huang, Y. (2019). *Automated Invoice Processing in International Trade*.
2. Heizer, J., Render, B., & Munson, C. (2020). *Operations Management: Sustainability and Supply Chain Management*.
3. Kotler, P., & Keller, K. L. (2021). *Marketing Management*.
4. Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems*.
5. Nielsen, J. (1994). *Usability Engineering*.
6. Stauffer, T. (2022). *Laravel: Up and Running*.
7. Sulaiman, F. (2022). *High-Performance Laravel with Octane*.
8. Wicaksono, A. (2023). *Implementasi Filament untuk Sistem Logistik Enterprise*.