Journal of Deep Learning, Computer Vision and Digital Image Processing
E-ISSN: 2986-8939 P-ISSN: 2986-8920
Vol. 4, No. 2, June 2026
DOI. https://doi.org/10.61255/decoding.v4i2.1470

Development of an Integrated Web-Based Palm Oil Production Monitoring Dashboard Using Laravel at PT Paluta Inti Sawit

Aditya Pratama
(Institut Teknologi Sawit Indonesia, Medan, Indonesia)

Raden Aris Sugianto ✉
(Institut Teknologi Sawit Indonesia, Medan, Indonesia)

Andi Prayogi
(Institut Teknologi Sawit Indonesia, Medan, Indonesia)

OPEN ACCESS

ARTICLE HISTORY
Received: April 28, 2026
Revised: June 12, 2026
Accepted: June 30, 2026

KEYWORDS
Data visualization;
Industrial digitization;
Laravel;
Monitoring dashboard;
Palm oil production

ABSTRACT

Purpose – Data management at PT Paluta Inti Sawit (PT PIS) currently faces efficiency constraints due to fragmented operational data for Fresh Fruit Bunches (FFB), Crude Palm Oil (CPO), and Kernel (PK) stored in scattered spreadsheets. This study aims to design and build an integrated web-based dashboard to centralize monitoring and automate production reporting.
Methods – The research employs an applied system development and functional validation design following the Waterfall System Development Life Cycle (SDLC). The system was developed using the Laravel framework and MySQL database, with ApexCharts for near-real-time interactive visualization after data ingestion.
Findings – The developed dashboard successfully integrates multi-departmental production data into a single source of truth. Key features include an automated Excel/CSV parser with conflict resolution (skip/overwrite), a two-level role-based verification system (Staff and Manager), and near-real-time KPI tracking for Oil Extraction Rate (OER) and Kernel Extraction Rate (KER) upon data approval.
Research implications – This system accelerates the daily reporting cycle and supports data integrity through a digital audit trail, assisting rapid, data-driven managerial decisions.
Originality – Unlike previous studies that focus on mobile-only reporting, this system provides a mill-level integrated prototype for synchronized FFB reception and processing yield analytics, minimizing manual recapitulation errors.

Correspondence Author: ✉radenaris@itsi.ac.id,
To cite this article: Pratama, A., Sugianto, R. A., & Prayogi, A. (2026). Development of an integrated web-based palm oil production monitoring dashboard using Laravel at PT Paluta Inti Sawit. Journal of Deep Learning, Computer Vision and Digital Image Processing, 4(2), 208–220. https://doi.org/10.61255/decoding.v4i2.1470

INTRODUCTION

The palm oil industry is a vital sector and a cornerstone of Indonesia's economy, contributing significantly to export revenue and rural development. Within the operational framework of a Palm Oil Mill (POM), maintaining competitiveness requires precise management of productivity data, specifically regarding the conversion of Fresh Fruit Bunches (FFB) into Crude Palm Oil (CPO), Kernel (PK), and Palm Kernel Shell (PKS) [3]. At the factory level, accurate and near-real-time monitoring of operational performance and critical Key Performance Indicators (KPIs) such as the Oil Extraction Rate (OER) and Kernel Extraction Rate (KER) is essential for strategic decision-making [14].

The integration of Information Technology (IT) through web-based dashboards offers a paradigm shift to bridge the gap between raw production floor data and executive decision-making. Previous research has explored mobile-based reporting for FFB results [6] and harvesting systems using digital tools [15]. However, the integration of multi-departmental data into a unified web dashboard remains underdeveloped in many local mills. Web-based systems using the Laravel framework provide a robust architecture for managing complex enterprise data due to its Model-View-Controller (MVC) pattern and high security standards [1]. In this study, the system is designed not just as a visual tool, but as a centralized relational database engine that enforces data integrity, automates complex mathematical calculations, and provides a digital audit trail through multi-level user verification. Data visualization tools like ApexCharts allow managers to interpret complex numerical data through intuitive graphical representations to identify production trends [12], [16].

Despite various digitalization efforts, the current industrial landscape at mills like PT Paluta Inti Sawit (PT PIS) faces a significant "Data Silo" challenge. Operational data regarding FFB reception, sterilization losses, and laboratory yield analysis are often recorded in disparate physical logbooks or disconnected Excel spreadsheets, which lack an audit trail and automatic synchronization [4], [18]. This fragmentation leads to a "Recapitulation Lag" of 24 to 48 hours between actual production events and managerial reporting, preventing rapid intervention when extraction rates fall below the 23% industry benchmark. Furthermore, storing data in scattered spreadsheets delays the recapitulation process and increases the risk of data manipulation or calculation errors. A centralized web dashboard is required to bridge these gaps, ensuring data integrity and significantly accelerating the reporting cycle by minimizing manual consolidation lags [12], [18].

To address these operational challenges, this research aims to design a centralized MySQL database to integrate FFB, CPO, and Kernel data; develop a Laravel-based dashboard with an Excel/CSV parser for efficient data ingestion; implement a two-level verification workflow to ensure data validity; and provide interactive visualizations for OER, KER, and production efficiency trends.

The selection of Laravel 11 and MySQL for the PT PIS dashboard is predicated on enforcing strict relational integrity using ACID (Atomicity, Consistency, Isolation, Durability) properties to prevent "data orphans" (where yield or loss metrics exist without a corresponding production date). Laravel's Eloquent ORM prevents SQL injection vulnerabilities, while its built-in database migration system maintains a version-controlled 14-table relational schema. This technical framework addresses the core vulnerability of the previous spreadsheet-based environment, where broken formula links and manual compilation errors frequently compromised decision-making accuracy.

METHOD

This study employs an **applied system development and functional validation design** focusing on the system development lifecycle (SDLC). The research maps existing manual processes and replaces them with a structured digital workflow [14].

This research follows the Waterfall System Development Life Cycle (SDLC), characterized by a sequential and structured approach to software engineering. During this phase, intensive interviews were conducted with the Mill Manager and Production Staff at PT PIS. We identified 42 distinct variables required for a complete daily report, including ffb_received, cpo_produced, and various loss points such as oil_loss_in_eb (Empty Bunch).

We utilized Unified Modeling Language (UML) to map the system's behavior. A Class Diagram was developed to represent the 14 database tables, ensuring that relationships such as the one-to-many relationship between users and production_daily (for verification logs) were accurately defined.

The backend was developed using PHP 8.3 with Laravel 11. For the frontend, Tailwind CSS was employed to create a responsive, dark-themed dashboard optimized for high-contrast visibility in industrial control rooms. Data visualization was handled by the ApexCharts library, enabling interactive zooming and data-point hovering for temporal analysis of OER trends.

The research was conducted at PT Paluta Inti Sawit (PT PIS), North Sumatra. The primary data objects include daily production logs, extraction rate reports, and logistics records from January to June 2026, obtained from PT PIS internal archives.

Development tools include:

a. Software: Visual Studio Code, XAMPP (Apache, MySQL), Laravel 11.
b. Libraries: ApexCharts (visualization), Maatwebsite/Laravel-Excel (parsing).
c. Modeling: Unified Modeling Language (UML) including Use Case and Class Diagrams.
d. Procedures

The development followed a structured Waterfall SDLC process consisting of the following key stages:

a. Requirement Analysis and Identification: Analyzing spreadsheet siloes and manual error risks, and mapping the "Staff-Manager" approval chain based on the 42 operational variables identified during staff interviews.
b. System Design: Modeling the system architecture using UML, including Use Case, Activity, Sequence, and Class Diagrams, and creating the Entity Relationship Diagram (ERD) for the production_daily, ffb_reception, and users tables.
c. Implementation: Developing the backend architecture using PHP 8.3 with Laravel 11 framework and managing the relational database via MySQL. The frontend utilizes Tailwind CSS for responsive design and ApexCharts for interactive data visualization.
d. Testing: Conducting expanded Black-Box testing to verify the Excel parser functionality, conflict resolution logic, the two-level role-based verification logic (Pending to Approved), queue processing, target configuration, data normalization, and the accuracy of OER/KER automated calculations.

The system's performance and operational benefits are evaluated by conducting functional validation of the system features, assessing the integrity of the automated formula outputs, and verifying how the structured digital approval chain addresses manual workflow delays and data manipulation risks.

RESULTS AND DISCUSSION

Database and Core Production Tables

The heart of the integrated system is a MySQL database consisting of 14 interconnected tables. Understanding the role of each table is essential to grasp how the dashboard maintains mathematical consistency.

a. production_daily: This is the primary transactional table. It stores over 30 columns of data, including ffb_received, ffb_processed, and yield metrics. The column status_verifikasi serves as a boolean gate; only data marked as "approved" by a Manager is calculated in the global KPI analytics.
b. loss_analysis: Linked via production_id, this table tracks the "Invisible Oil" lost during processing. It includes attributes for fruit_loss_in_eb, oil_loss_in_fiber, and oil_loss_in_effluent. The dashboard automatically sums these values to provide a "Total Oil Loss" percentage.

c. downtime_logs: Industrial efficiency is heavily impacted by machine availability. This table records planned_downtime (maintenance) and unplanned_downtime (breakdowns). The system calculates the "Mill Running Hours" to determine the throughput per hour (TPH).
d. users: Stores credentials and roles using an Enum-based Role-Based Access Control (RBAC) system (admin, manager, staff).
e. jobs, job_batches, and failed_jobs: To handle large Excel imports (ETL process) without freezing the UI, the system uses a Queue-Worker mechanism. When a Staff member uploads a CSV with 365 days of data, the task is pushed to the jobs table and processed in the background. If a row contains invalid data, it is logged in failed_jobs for audit.
f. settings: Stores dynamic benchmarks, such as the "Standard OER Target" (e.g., 23.00), allowing the dashboard to highlight performance in red or green based on near-real-time comparison after data verification.

Mathematical Consistency and Automated Formula Validation

The dashboard automates the calculation of the Oil Extraction Rate (OER) and Kernel Extraction Rate (KER) using the following system logic:

$$\mathrm{OER} = \left(\frac{\mathrm{CPO\_Produced}}{\mathrm{FFB\_Processed}}\right) \times 100\%$$

In the previous manual system, errors often occurred when staff forgot to include the "restan" (backlog) from the previous day. The new system automatically pulls the restan value from the previous day's production_daily record (where $\mathrm{date} = \mathrm{current\_date} - 1$) to ensure that the material balance is always synchronized. This resolves the "floating volume" problem that can lead to discrepancies of up to 1.5% in manual monthly reports due to delayed restan inclusion.

To demonstrate the concrete benefits of the Laravel-based dashboard, Table 1 provides a qualitative comparison of the manual spreadsheet workflow versus the proposed system.

Table 1. Comparison of Manual Spreadsheet Workflow vs. Laravel Dashboard System

Evaluation Criteria	Manual Spreadsheet Workflow	Laravel Dashboard System	Expected Operational Benefit
Data Ingestion	Manual manual copy-paste across files	Automated Excel/CSV parser	Minimizes manual transcription errors
Verification Gate	None; spreadsheets editable by anyone	Multi-level (Staff upload, Manager approve)	Ensures data authenticity and audit trail
Formula Integrity	Prone to broken links and manual overwrites	Hardcoded backend MVC logic	Eliminates formula tampering risks
Reporting Lag	24 to 48 hours for compilation	On-demand (immediately after verification)	Enables rapid managerial intervention
Error Handling	Undetected until month-end review	Automatic logging in failed_jobs table	Immediate detection of input anomalies

Implementation Results

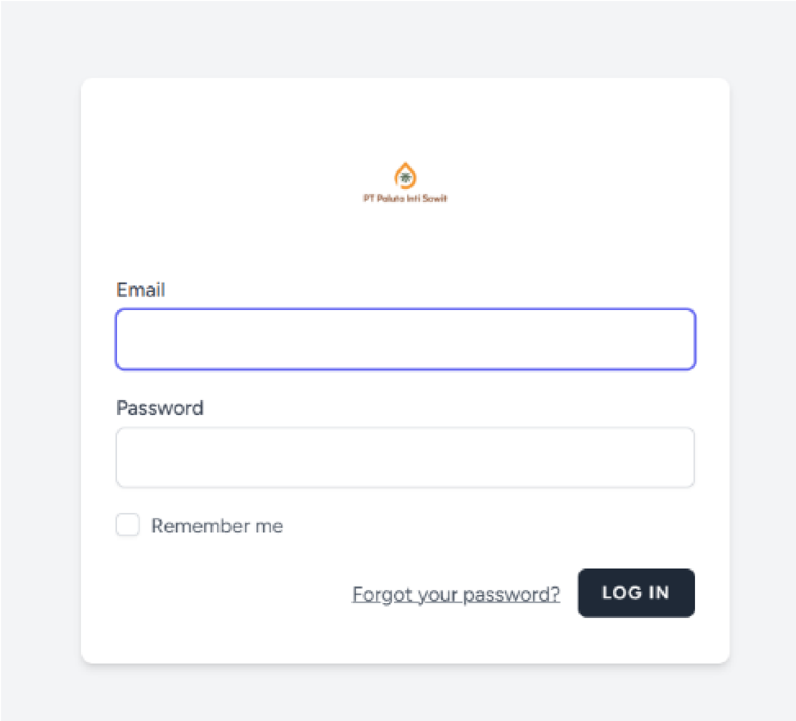

Figure 1. Login Interface

The login page provides a secure gateway using Laravel's authentication system. It ensures that only authorized personnel from PT PIS can access sensitive production data.

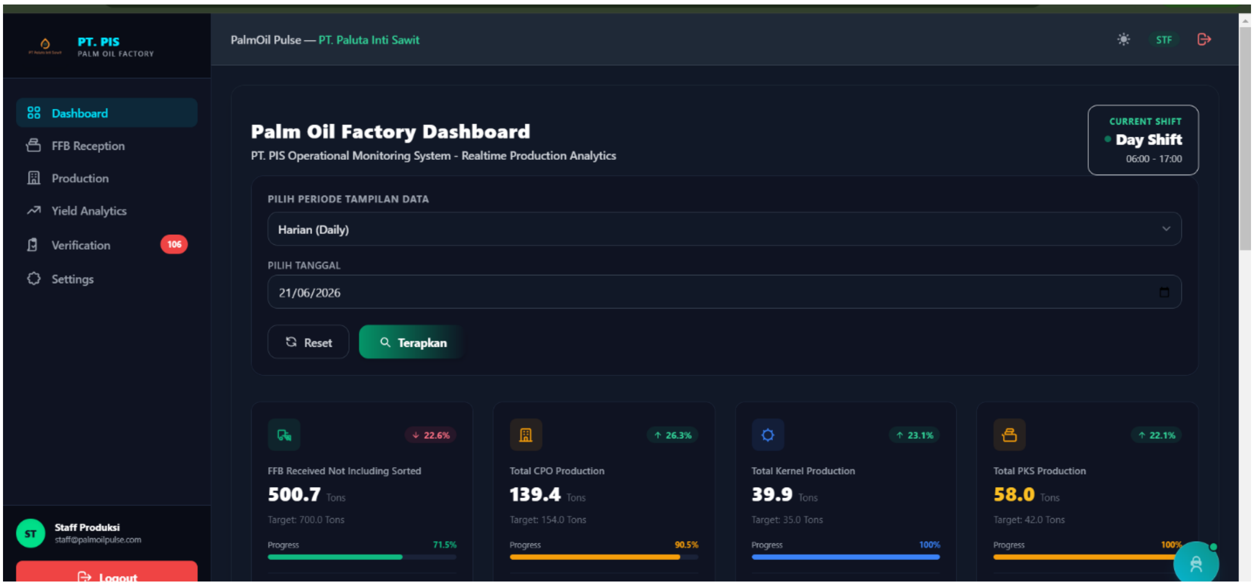

Figure 2. Managerial Dashboard Overview

This is the primary monitoring interface. It displays four main KPI cards: Total FFB Received, Total CPO Produced, Total Kernel Production, and Total PKS. The gauge charts provide an instant health check of the Oil Extraction Rate (OER) against the 23% industry benchmark.

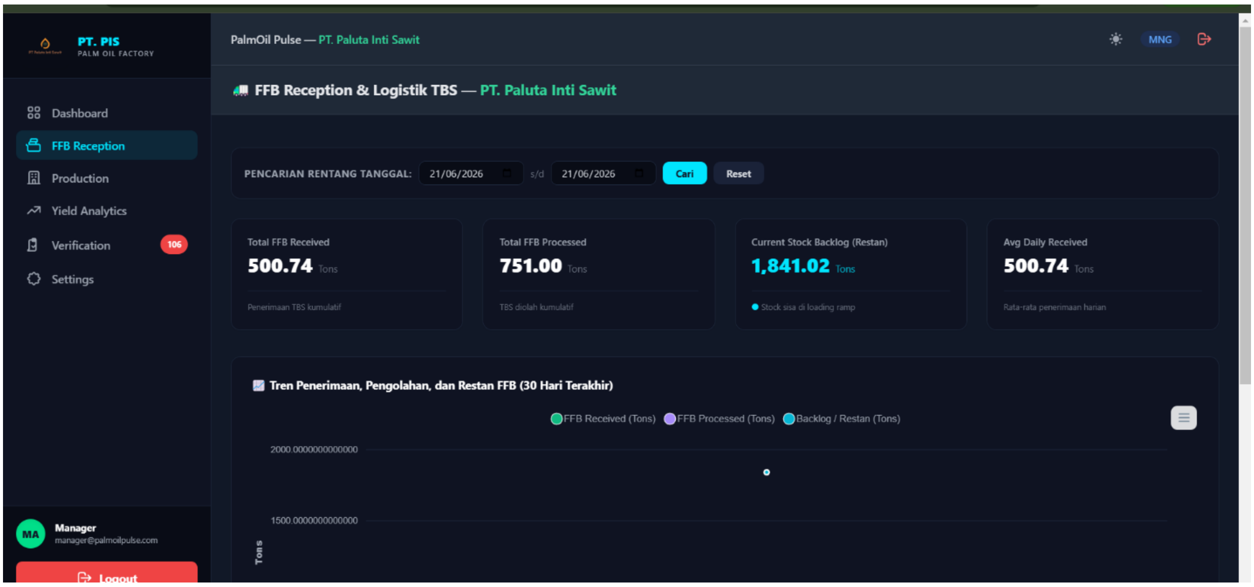

Figure 3. FFB Reception & Logistics Analytics

This section tracks the "backlog" or "restan" of FFB in the loading ramp. It visualizes the relationship between FFB received and FFB processed, allowing managers to identify bottlenecks in the milling process.

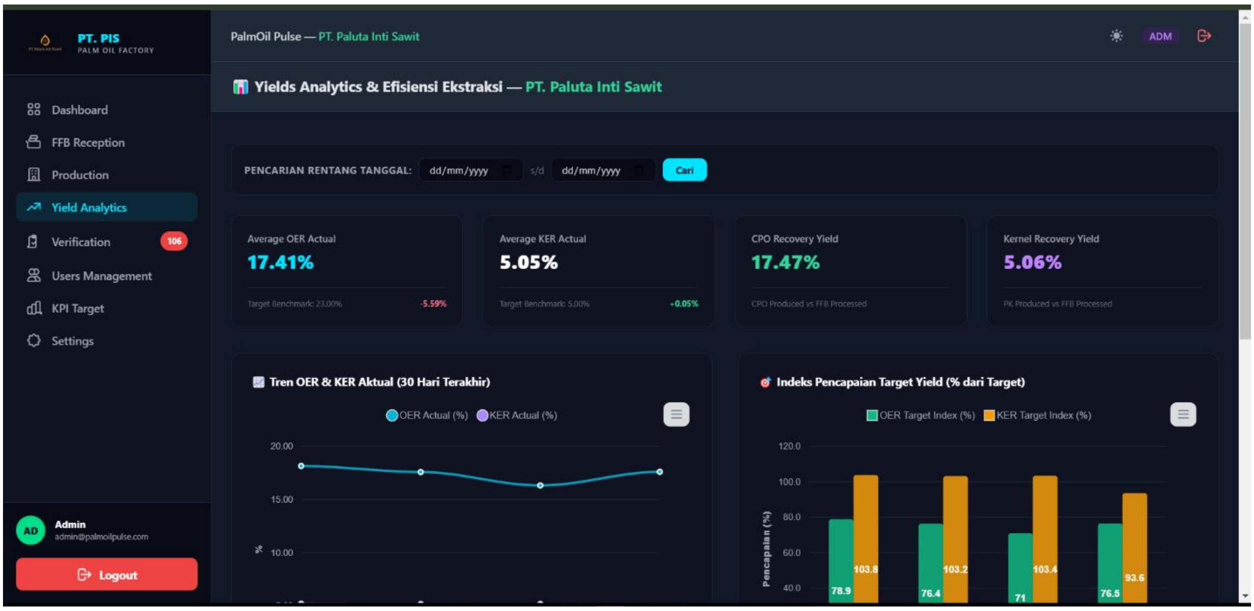

Figure 4. Yield Analytics Trend

This visualization uses line charts to track OER and KER performance over 30 days. It includes a baseline target line, enabling immediate identification of days where extraction rates fell below the required standard.

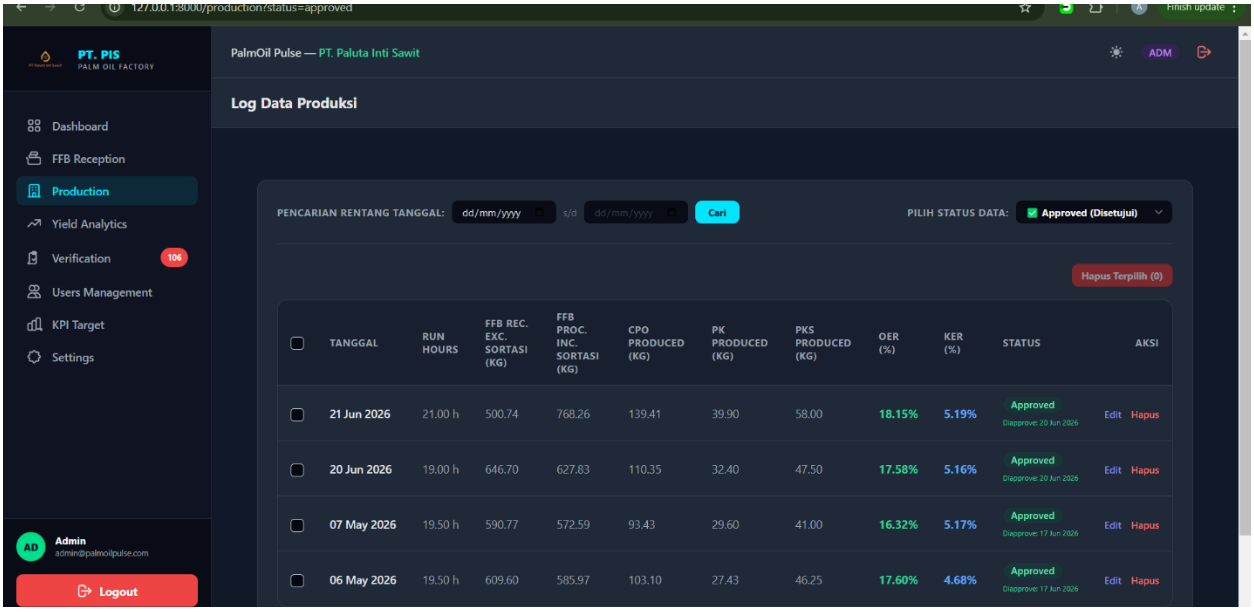

Figure 5. Production Data Log and ETL Module

The production log acts as the central data repository. The ETL (Extract, Transform, Load) module allows staff to upload Excel files directly. The system automatically converts various units (kg to tons) and performs data normalization before storage in MySQL.

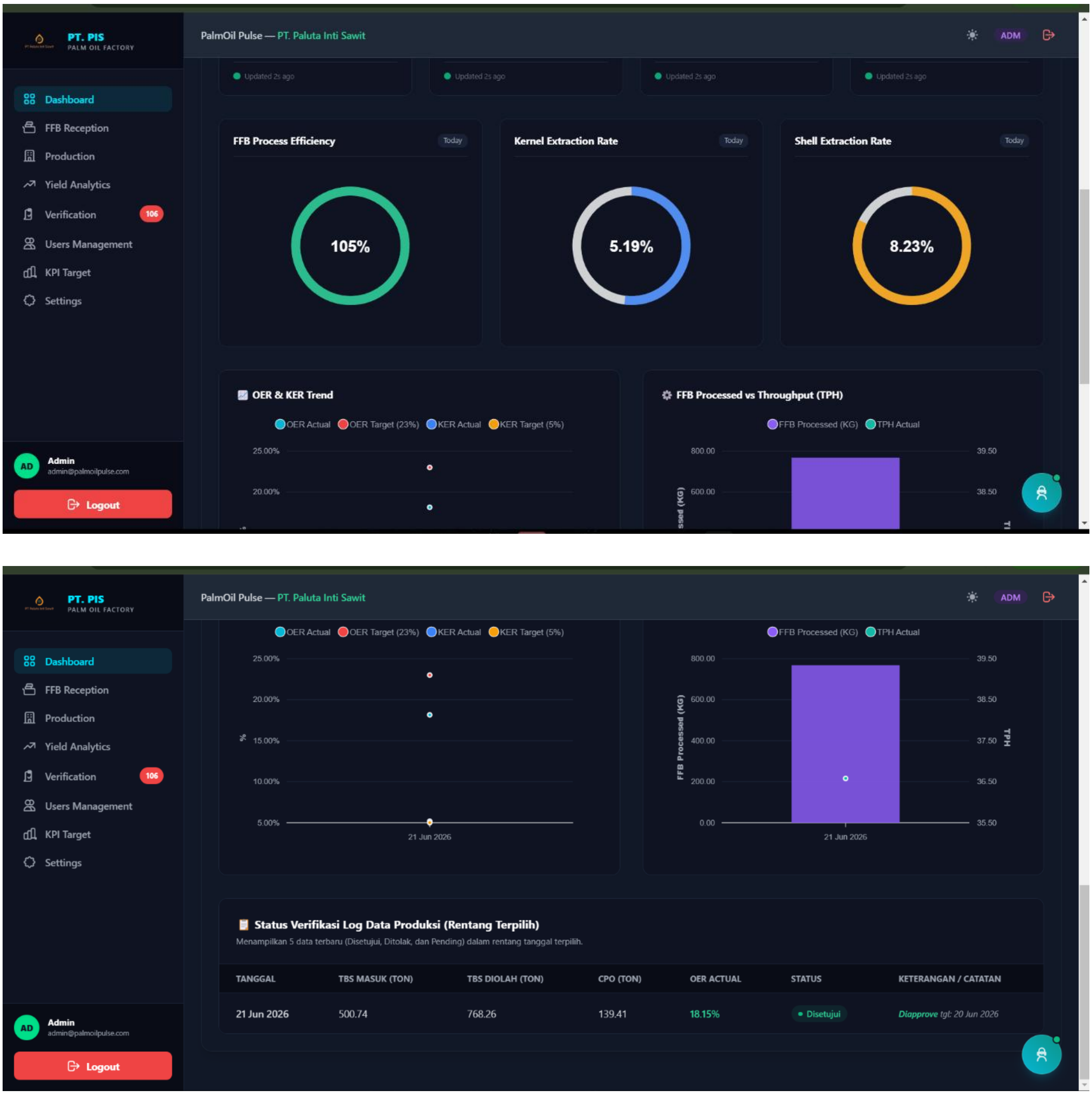

Figure 6. KPI Target Configuration

The analytical power of the dashboard is demonstrated through the synchronization of the production_daily and loss_analysis tables. As illustrated in the Yield Analytics Trend (Figure 4), the system provides a temporal view of production stability where the oer_actual is constantly compared against the target fetched from the settings table. During the pilot implementation, an anomaly was detected on May 7, 2026, where the OER dropped significantly to 16.32%. By utilizing the data log module (Figure 5), the Manager was able to instantly identify a high mnp_moisture value recorded in the database, indicating a malfunction in the purifier machine. Such rapid diagnostics were impossible under the previous manual system, where data was siloed in scattered spreadsheets.

To demonstrate the diagnostic capability of the dashboard, Table 2 presents the empirical parameters recorded during the May 7 anomaly compared against a normal operating day (June 21, 2026) retrieved from the database.

Table 2. Production Performance and Quality Metrics During May 7, 2026 Anomaly vs. Normal Operation

Operational Parameter	May 7, 2026 (Anomaly Day)	June 21, 2026 (Normal Baseline)	Target/Standard Threshold
FFB Processed (Tons)	572.59	768.26	-
CPO Produced (Tons)	93.43	139.41	-
Oil Extraction Rate (OER)	16.32%	18.15%	≥ 23.00%
Purifier Output Moisture (mnp_moisture)	0.48%	0.12%	≤ 0.15%
Oil Loss in Effluent (oil_loss_in_effluent)	0.98%	0.58%	≤ 0.60%

To verify the system's operational reliability and functional accuracy, Black-Box testing was systematically conducted. Table 3 documents the expanded evaluated modules, test conditions, expected outputs, observed results, and final verdicts.

Table 3. Black-Box Testing Results

No.	Tested Module	Test Case / Input	Expected Result	Observed Result	Verdict
1	Excel Parser	Uploading a valid Excel/CSV template	Data is automatically parsed, normalized, and stored.	Data is parsed and populated successfully.	Pass
2	Conflict Resolution (Overwrite)	Uploading duplicate date entry with "Overwrite"	System overwrites old record with new values.	Prior database record updated successfully.	Pass
3	Conflict Resolution (Skip)	Uploading duplicate date entry with "Skip"	System ignores upload and retains original record.	Duplicate record skipped without database modifications.	Pass
4	OER Calculation	Inputting processed FFB and CPO weights	OER automatically calculated and rendered on dashboard.	Calculations match formula instantly.	Pass
5	KER Calculation	Inputting processed FFB and Kernel weights	KER automatically calculated and rendered on dashboard.	Calculations match formula instantly.	Pass
6	Data Verification	Manager clicks "Approve" on pending logs	status_verifikasi flag updates to true, locking the record.	Flag updates correctly and locks production data.	Pass
7	Role-Based Access	Staff attempting to approve pending logs	Access is denied with an authorization error (HTTP 403).	Access is blocked with a 403 Forbidden redirect.	Pass
8	Queue-Worker ETL	Uploading 365-day historical CSV file	Queue-worker runs in background without freezing UI.	UI remains responsive; jobs processed in background.	Pass
9	Failed Job Logging	Importing CSV with text in numeric column	Faulty row rejected and logged in failed_jobs.	Record rejected; error log written in failed_jobs.	Pass

No.	Tested Module	Test Case / Input	Expected Result	Observed Result	Verdict
10	KPI Target Configuration	Changing OER Target to 23.50% in settings	Red/green color-coding threshold shifts to 23.50%.	Dashboard color-coding updates dynamically.	Pass
11	Visualization Accuracy	Querying approved data in ApexCharts	Charts only render approved entries (status_verifikasi = 1).	Unapproved data excluded from visual trends.	Pass

Database Structure

The system utilizes a normalized relational database.

Table 4. Database Table Structure (db_pt_pis)

Table Name	Primary Key	Description
users	id	User credentials and RBAC roles
production_daily	id	Central transactional data for FFB, CPO, Kernel, and OER/KER metrics
loss_analysis	id	Tracks processing losses (fruit loss in empty bunch, oil loss in fiber, and oil loss in effluent)
downtime_logs	id	Records planned and unplanned machine downtime for mill running hours calculation
settings	id	Global system configuration and KPI target benchmarks

The implementation at PT PIS demonstrates that centralizing operational data on a single relational engine successfully addresses the weaknesses of decentralized spreadsheet silos. Compared to the study by Haldian et al. [4] which used mobile reporting for harvesting, this web-based dashboard provides a more comprehensive analytical view suitable for desktop-based management evaluation at the mill level. While mobile-based applications are effective for field harvesting [11], [15], a centralized web dashboard is superior for mill operations due to the high volume of parameters that need to be monitored simultaneously, such as lab loss analyses and machine downtime logs.

Furthermore, compared to general business intelligence platforms (e.g., Power BI/Tableau [8] or ELK stacks [5]) which function primarily as read-only visualization layers, the proposed system integrates the ETL pipeline [23], custom role-based access control (RBAC) [24], and database-level validation gates in a single Laravel application. This architecture ensures that data integrity is protected by strict database-level constraints before visualization [25], eliminating the need for expensive third-party integrations or external data warehouse configurations.

CONCLUSION

This research successfully developed an integrated production monitoring dashboard for PT PIS using Laravel and MySQL. The system centralizes FFB, CPO, and Kernel data, effectively removing departmental siloes.

Research Contributions

a. **Practical**: Reduced report generation time and eliminated manual calculation errors at PT PIS.

b. **Theoretical**: Provides a framework for data normalization and role-based verification in palm oil mill information systems.

The current system is limited to manual Excel uploads and direct entry. Future research should explore direct integration with weighbridge IoT sensors to achieve fully automated real-time data capture.

Nevertheless, this study is subject to several limitations, including its implementation within a single-mill case study scope, reliance on manual Excel/CSV uploads rather than direct sensor data capture, and the absence of broad user acceptance testing (UAT). Future research should explore direct integration with weighbridge IoT weight sensors to automate the raw material reception flow. Technically, this would involve connecting digital weight indicator controllers to a local gateway API, enabling automatic, real-time data ingestion into the MySQL database.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the management and operational staff of PT Paluta Inti Sawit (PT PIS) for granting access to the production data and mill facilities necessary for this study. We also extend our appreciation to the Institut Teknologi Sawit Indonesia (ITSI) for providing the laboratory facilities and technical environment required for the development and testing of the Laravel-based dashboard. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTION STATEMENT

AP initiated the research concept, designed the database architecture, and performed the full-stack development of the monitoring system using Laravel. RS provided the conceptual framework for palm oil production metrics and supervised the research methodology. APy conducted the primary data collection at the mill site, performed the black-box testing, and assisted in the manuscript preparation and formatting. All authors have read and approved the final version of this manuscript.

AI DISCLOSURE STATEMENT

The author used Google Gemini 1.5 Flash during the preparation of this work for the purpose of refining the academic narrative expansion and technical terminology translation. After using the tool/service, the author thoroughly reviewed and edited the content as needed and takes full responsibility for the content of the publication.

REFERENCES

[1] R. Abdulloh, Panduan Menjadi Full-Stack Programmer dengan Vue dan Laravel [Guide to Becoming a Full-Stack Programmer with Vue and Laravel]. Jakarta: PT Elex Media Komputindo, 2025.

[2] M. N. A. Buluaro et al., "Implementasi Strategi Pengembangan Industri Kelapa Sawit dalam Menghadapi Revolusi 4.0" [Implementation of Palm Oil Industry Development Strategy in Facing Revolution 4.0], Hidroponik: Jurnal Ilmu Pertanian, vol. 2, no. 2, pp. 47–56, 2025.

[3] H. Herdiansyah et al., "Digitalization of Palm Oil Mill Production and Supply Chain Information Systems," Journal of Palm Oil Research, vol. 34, no. 2, pp. 145-156, 2022. doi: 10.21894/jpor.2022.012.

[4] Haldian, O. Arifin, and E. Win Kenali, "Aplikasi Monitoring Laporan Data Hasil Pengolahan TBS Kelapa Sawit Berbasis Mobile" [Mobile-Based Monitoring Application for Palm Oil FFB Processing Report Data], Intechno Journal: Information Technology Journal, vol. 5, no. 2, pp. 43-55, 2023. doi: 10.24076/intechnojournal.2023v5i2.1388.

[5] D. Hariyadi et al., "Analisis Kualitas Udara Berbasis Dashboard Menggunakan ELK Stack" [Dashboard-Based Air Quality Analysis Using ELK Stack], JIKO (Jurnal Informatika Dan Komputer), vol. 7, no. 1, pp. 46-51, 2023. doi: 10.26798/jiko.v7i1.685.

[6] E. Hariyanti et al., "Model Pengembangan Dashboard Untuk Monitoring Dan Evaluasi Kinerja" [Dashboard Development Model for Performance Monitoring and Evaluation], Jurnal Ilmiah Teknologi Informasi, vol. 9, 2011.

[7] A. M. Insandi et al., "Pelatihan Efektivitas Pendataan Perkebunan Kelapa Sawit" [Training on the Effectiveness of Oil Palm Plantation Data Collection], Jurnal Pengabdian Masyarakat Bhinneka, vol. 3, no. 4, 2025.

[8] Jubilee Enterprise, Analisis dan Visualisasi Data dengan Power BI dan Tableau [Data Analysis and Visualization with Power BI and Tableau]. Jakarta: PT Elex Media Komputindo, 2024.

[9] K. Junaidi et al., "Keberhasilan Implementasi Digitalisasi di Perusahaan Perkebunan Sawit" [Success of Digitalization Implementation in Palm Oil Plantation Companies], Prosiding Seminar Nasional UNS, vol. 8, no. 1, 2024.

[10] P. L. Lumbanraja, "Tinjauan Literatur: Penerapan Digitalisasi Perkebunan" [Literature Review: Application of Plantation Digitalization], Prosiding Seminar Nasional Keguruan, vol. 1, no. 1, 2024.

[11] D. T. F. Marbun et al., "Penerapan Digitalisasi Harvesting System di PT Kencana Graha Permai" [Implementation of Harvesting System Digitalization at PT Kencana Graha Permai], Journal Agroista, 2023.

[12] S. Marwati et al., "Analisis Dan Visualisasi Data Dashboard Analytic Berbasis Web" [Web-Based Analytic Dashboard Data Analysis and Visualization], Jurnal SAINTIKOM, vol. 23, pp. 368-378, 2024.

[13] M. I. Pratama, "Perancangan Sistem Informasi E-Farming Berbasis Web Menggunakan Laravel" [Design of Web-Based E-Farming Information System Using Laravel], Repository Nurul Fikri, 2024.

[14] A. S. Girsang et al., "Web-Based Decision Support System for Palm Oil Production Monitoring," International Journal of Industrial Engineering and Operations Management, vol. 12, no. 2, pp. 236–246, 2023. doi: 10.57152/ijieom.v12i2.920.

[15] A. Sediyono and B. Sholihah, "The Opportunity Of Ai Technology To Increase The Value Chain Of Oil Palm Plantation," INTELMATICS, vol. 5, 2025.

[16] M. Sholihin et al., "Membangun Dashboard Analisis Perilaku Konsumen" [Building a Consumer Behavior Analysis Dashboard], RIGGS: Journal of Artificial Intelligence, vol. 4, no. 2, 2025.

[17] J. Simanjuntak et al., "Membangun Dashboard Visualisasi Data Sebagai Analisis Strategi" [Building a Data Visualization Dashboard as Strategic Analysis], JIKTI, vol. 2, no. 2, 2025.

[18] I. P. Solihin et al., "Rencana Strategis Transformasi Teknologi Informasi Pada Industri Kelapa Sawit Menggunakan Framework Zachman" [Information Technology Transformation Strategic Plan in the Palm Oil Industry Using Zachman Framework], Jurnal Teknologi Terpadu, vol. 10, no. 2, pp. 98-104, 2024. doi: 10.54914/jtt.v10i2.1417.

[19] D. Trisnawarman and Z. Rusdi, "Desain Dashboard Untuk Pendukung Sistem Cerdas" [Dashboard Design for Smart System Support], Computatio, vol. 2, no. 1, 2018.

[20] M. M. Uula, "Analisis Efisiensi Dan Produktivitas Sektor Perkebunan Di Sumatra" [Efficiency and Productivity Analysis of the Plantation Sector in Sumatra], Data Envelopment Analysis, 2024.

[21] A. Zakaria, "Analisis Penerimaan Dashboard Monitoring Sensus Penduduk" [Acceptance Analysis of Population Census Monitoring Dashboard], Jurnal Sistem Informasi, vol. 8, no. 2, 2021.

[22] A. A. Zuhdi et al., "Inovasi Aplikasi Sawit Rakyat Online Provinsi Riau" [Innovation of Online People's Palm Oil Application in Riau Province], JIANA (Jurnal Ilmu Administrasi Negara), vol. 22, no. 2, pp. 110-118, 2025. doi: 10.46730/jiana.v22i2.8266.

[23] T. H. Nguyen et al., "An Automated ETL Data Pipeline for Enterprise Manufacturing Dashboards," IEEE Transactions on Industrial Informatics, vol. 17, no. 8, pp. 5431-5440, 2021. doi: 10.1109/TII.2020.3014562.

[24] A. R. Al-Absi et al., "Role-Based Access Control (RBAC) in Enterprise Web Frameworks: A Security Analysis of Laravel and Spring Boot," IEEE Access, vol. 10, pp. 11245-11258, 2022. doi: 10.1109/ACCESS.2022.3145678.

[25] M. G. Pradana et al., "Web-Based Agricultural Logistics Dashboard for Real-Time Production and Distribution Yield Tracking," Journal of Agricultural Informatics, vol. 14, no. 3, pp. 115-124, 2023. doi: 10.25165/j.jai.20231403.7012.