

Development and Evaluation of a No-Code Key Performance Indicator (KPI) Dashboard Using Base44 for Divisional Performance Monitoring (Case Study: Macwish Indonesia)

Amalia^{1)*}, Muhammad Akhyar Abdan Syakuron²⁾, Amalia Rahmah³⁾

¹⁾Information Systems, ²⁾Digital Business, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, Indonesia

¹⁾amalia@nurulfikri.ac.id, ²⁾muha22007bd@student.nurulfikri.ac.id, ³⁾amalia.rahmah@nurulfikri.ac.id

ABSTRACT

Divisional performance monitoring that still relies on spreadsheets prevents key performance indicator (KPI) achievements from being presented in a centralized manner, requires manual data processing, and slows down evaluation and decision-making. This condition was found at Macwish Indonesia, a macaroni-based food and beverage business that originated from a campus entrepreneurship program. This study aims to develop and evaluate a no-code KPI dashboard built on the Base44 platform to monitor divisional performance. The study employed a Research and Development (R&D) approach with the Waterfall model, comprising problem identification, literature review, data collection, design and implementation, testing, and evaluation. User requirements were elicited through observation and semi-structured interviews with three division heads. The resulting dashboard provides login, division and KPI management, performance input, a monitoring dashboard, and quarterly evaluation features. Testing was conducted using Black Box Testing and User Acceptance Testing (UAT) with a four-point Likert scale. All 18 Black Box Testing items passed in the functional scenarios tested, which did not cover access rights, data security, or system resilience. The UAT, involving three division heads and one expert practitioner, yielded scores of 96.25% for system functionality, 97.50% for ease of use, 98.75% for performance and reliability, and 96.25% for user satisfaction, with an overall mean of 97.19% ("Very Good" category). These results are an initial indication of user acceptance in a single case study with four respondents and therefore cannot be generalized, and the advantage over spreadsheets was not measured quantitatively. This study has so far demonstrated only functional feasibility and initial user acceptance, not an improvement in monitoring effectiveness. It contributes to the literature on the use of no-code platforms for performance monitoring systems and presents a replicable implementation example for small and medium-sized food and beverage businesses.

Keywords: Dashboard; Key Performance Indicator; No-Code; Base44; Performance Monitoring

INTRODUCTION

The food and beverage (F&B) industry is a competitive and dynamic sector. Its operational success depends on service speed, inventory management, and the ability to adapt to customer demand. To support this, organizations require performance management, that is, the planning, measurement, monitoring, and evaluation of work results so that organizational goals are achieved in a structured manner. Performance measurement that relies solely on financial indicators is considered inadequate; the balanced scorecard approach therefore views performance from several perspectives, including the financial, customer, and internal process perspectives, and this approach continues to be developed in recent studies (Silva et al., 2025). In practice, these measures are formulated as key performance indicators (KPIs) that can be applied at the divisional level.

Macwish Indonesia is a brand under PT Lokal Bisa Mendunia that offers macaroni-based fast food and originated from a campus entrepreneurship program. The organization is led by a Chief Executive Officer

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

(CEO) and has four divisions: finance, marketing, customer, and operations. Interviews with three division heads revealed that KPI monitoring is still carried out using spreadsheets. Each division records its KPI achievements periodically, but these data cannot be used directly as monitoring information because the calculation of achievements, recapitulation, and preparation of visualizations are still performed manually. All three respondents stated that no real-time KPI monitoring system is available. Management must wait until all data have been entered and processed before it can view divisional performance, and this process is prone to data-entry and formula errors.

A dashboard is a visual display of consolidated key information that allows performance to be monitored quickly (Arputharaj et al., 2024). The role of dashboards in performance management continues to be studied and applied across various organizations (Arputharaj et al., 2024; Nunes et al., 2024). Their development does not necessarily require conventional programming. Low-code/no-code platforms enable applications to be built through visual interfaces or textual instructions, thereby lowering the programming expertise required and allowing greater involvement of non-technical users (Ajimati et al., 2025; Hagel et al., 2024). Recent systematic reviews indicate that the adoption of low-code/no-code is growing as a means of citizen development for digital transformation (Ajimati et al., 2025) and has begun to be used to digitalize management processes in small and medium-sized enterprises (SMEs) (Domański et al., 2023).

Several studies in Indonesia have developed KPI-based performance monitoring systems. Kristofher et al. (2022) designed an employee performance monitoring dashboard using a User-Centered Design approach, but their evaluation was limited to design validation without a functional prototype. Aryanto and Susila (2024) built a KPI visualization dashboard for an IT helpdesk team using Looker Data Studio, but the testing reported was limited to Black Box Testing. Fakhruddin (2025) developed a web-based employee KPI appraisal application using the Laravel and Vue.js frameworks, while Yoewono et al. (2024) designed an employee discipline dashboard using Grafana. None of these studies built a dashboard using a no-code platform based on text descriptions, nor did they test user acceptance of a divisional KPI monitoring system in a small-scale food and beverage business.

At the international level, Nunes et al. (2024) implemented KPIs and Power BI dashboards in an automotive components company, whereas Bennett and Müller Loose (2024) developed an online KPI dashboard for SMEs in the wine sector together with web designers and programmers. Neither study addressed the development of a divisional KPI dashboard using a text-description-based no-code platform in a small-scale F&B business.

Based on these conditions, this study aims to develop a no-code KPI dashboard using the Base44 platform for divisional performance monitoring at Macwish Indonesia and to evaluate it in terms of functional conformity and user acceptance. The contributions of this study are: (1) a no-code KPI dashboard that can be developed without manual programming, (2) an implementation in the real-world context of a student-run business in the F&B sector, and (3) an evaluation using Black Box Testing and User Acceptance Testing (UAT), together with a discussion of its limitations.

LITERATURE REVIEW

Based on the literature review, related studies are grouped into three main themes:

1. Performance management, KPIs, and monitoring dashboards
2. No-code/low-code platforms and digital transformation
3. System development and evaluation methods

This mapping clarifies the position of the study and the research gap. Previous studies have extensively examined performance dashboards and low-code platforms separately, but the application of no-code to divisional KPI dashboards that are implemented and tested for user acceptance in small-scale F&B businesses has rarely been discussed.

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Performance Management, KPIs, and Dashboards

Performance management encompasses the continuous planning, measurement, monitoring, and evaluation of performance. A KPI is a measurable metric used to assess the achievement of individuals or divisions against established objectives. The idea of measuring performance from multiple perspectives (Silva et al., 2025) is consistent with Macwish's division into finance, customer, marketing, and operations divisions. To be informative, KPIs are generally formulated according to the SMART principle, namely specific, measurable, achievable, relevant, and time-bound.

Dashboards present KPIs in concise charts and tables so that performance can be read at a glance (Arputharaj et al., 2024). Arputharaj et al. (2024) combined the Analytic Hierarchy Process (AHP) and the objective matrix with a balanced scorecard model in a dashboard that visually presents performance progress over time. In Indonesia, Aryanto and Susila (2024) developed a KPI dashboard using Looker Data Studio, but the testing reported was limited to Black Box Testing. Other international studies show a similar tendency. Nunes et al. (2024) implemented 22 new cross-departmental KPIs in an automotive components company and presented them through a Power BI dashboard, enabling senior managers to analyze achievements in greater detail and make decisions faster. Bennett and Müller Loose (2024) developed a KPI-based online dashboard for SMEs in the wine sector through a user-centered development process that included 24 requirement-elicitation interviews and usability tests with 10 participants. Both studies underline the importance of involving users in requirement elicitation and dashboard evaluation, but neither discusses building a dashboard with a text-description-based no-code platform.

No-Code/Low-Code Platforms and Digital Transformation

Digital transformation in SMEs encompasses the use of digital technology, strategy, and digital capabilities in organizational and management processes (Marino-Romero et al., 2024). Low-code platforms are one of its enablers because they allow applications to be built through visual models and configuration with little or no hand-written code (Ajimati et al., 2025). Domański et al. (2023) reviewed low-code/no-code platforms for the digitalization of management processes in SMEs, while Ajimati et al. (2025) mapped the benefits and challenges of their adoption in citizen development practice.

Base44 is a no-code platform that builds applications from text descriptions, so that user requirements, performance indicators, and monitoring functions can be translated into data structures and system interfaces. This approach is consistent with recent research. Hagel et al. (2024) showed that large language models (LLMs) can replace text-based modeling in low-code platforms, bringing such platforms closer to true no-code for non-professional users. Domański et al. (2023) noted that studies specifically addressing these platforms in SMEs remain very limited. Existing low-code/no-code studies mostly focus on general platform reviews and platform rankings. Their application to divisional KPI dashboards in small-scale F&B businesses has not been widely documented.

System Development and Evaluation Methods

The Waterfall model structures software development sequentially, from requirements and design through implementation to testing (Saravanos, 2025), so that each stage is clearly documented. The Research and Development (R&D) approach is used to produce a product while testing its effectiveness (Husnayayin et al., 2024).

In evaluation, Black Box Testing verifies system outputs based on inputs without examining the code structure, whereas UAT is carried out by end users to assess whether the system meets their needs and is fit for use. The Technology Acceptance Model (TAM) positions perceived usefulness and perceived ease of use as the main determinants of user acceptance (Mogaji et al., 2024), and in this study it is used only as a conceptual foundation. The UAT instrument consists of four system evaluation aspects (system

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

functionality, ease of use, performance and reliability, and user satisfaction) and does not measure the full set of TAM constructs, such as attitude toward use, behavioral intention, and actual use. This study therefore does not empirically test the TAM. A Likert scale was used to measure respondents' perceptions of these four aspects.

Research Positioning

Table 1 compares previous studies with the present study.

Table 1. Research Positioning

No	Researcher	Research Focus	Approach	Limitation	Position of This Study
1	Aryanto & Susila (2024)	KPI visualization dashboard for an IT helpdesk team	Looker Data Studio with Google Sheets data	Reported testing limited to Black Box Testing	Dashboard implemented and tested with Black Box Testing and UAT
2	Fakhruddin (2025)	Web-based employee KPI appraisal application	Laravel and Vue.js	Code-based development	No-code development with Base44
3	Yoewono et al. (2024)	Employee discipline dashboard	Grafana and Solo Scrum	Focuses on discipline rather than divisional KPIs	Focuses on divisional KPIs
4	Nunes et al. (2024)	KPI and dashboard implementation in an automotive components company	Power BI dashboard with input from cross-departmental employees	Focuses on automotive manufacturing; does not address text-description-based no-code	Divisional KPI dashboard with no-code in an F&B business
5	Bennett & Müller Loose (2024)	Online economic sustainability dashboard for SMEs in the wine sector	User-centered development: interviews and usability tests	Developed with web designers and programmers; not no-code	No-code development with Base44 and UAT
6	Domański et al. (2023)	Review of low-code/no-code platforms for SME management digitalization	Systematic literature review and grey incidence analysis	Platform review and ranking; not a KPI dashboard	Implementation of a KPI dashboard with no-code
7	This study	Divisional KPI dashboard	Waterfall, Base44, Black Box Testing, UAT	-	No-code, divisional KPIs, functional and acceptance testing

Based on this comparison, this study is positioned as the development of a no-code divisional KPI dashboard that is implemented and tested with a real-world partner.

2.5 Novelty and Contribution of the Study

The literature analysis shows that a gap remains in the development of KPI monitoring systems that (1) adopt a no-code approach, (2) target small-scale F&B businesses, (3) are easy to use for non-technical users, and (4) are evaluated in terms of function and user acceptance.

The novelty of this study is contextual and implementation-oriented rather than methodological. Base44 is a commercially available product; the scientific contribution of the study therefore lies not in a new algorithm or method, but in the *application novelty* of a no-code approach to divisional KPI dashboards in a context that has received little attention, together with documentation of the design and evaluation results.

* Corresponding author



Its contribution is thus closer to that of a technology application case study.

3. RESEARCH METHOD

Research Design

This study adopted a Research and Development (R&D) approach with the Waterfall development model. R&D was chosen because the study does not merely analyze a problem but also produces a no-code KPI dashboard whose feasibility and acceptance are tested (Husnayayin et al., 2024). The Waterfall model was chosen because of its sequential, documented flow (Saravanas, 2025) and because the system requirements could be identified at the outset through interviews with users. The maintenance stage of the Waterfall model is outside the scope of this study. The research stages were as follows:

1. **Problem identification:** initial observation of the divisional KPI measurement and monitoring process at Macwish Indonesia.
2. **Literature review:** examination of journals, books, and documents related to information systems, KPIs, dashboards, and no-code technology.
3. **Data collection:** semi-structured interviews with division heads to explore the existing process, constraints, and system requirements.
4. **Design and implementation:** specification of actors, system requirements, use cases, activity diagrams, and the data model, followed by implementation of the dashboard on Base44.
5. **Testing:** Black Box Testing for system functions and UAT for user acceptance.
6. **Evaluation:** assessment of system feasibility based on the test results and formulation of conclusions and recommendations.

The study was conducted at the Macwish Indonesia outlet located in the canteen of Campus A, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, Depok, West Java. The study limited the dashboard to the KPIs of four divisions (finance, marketing, customer, and operations), the KPI entry and monitoring process, and internal use. The determination of KPIs is entirely the authority of the CEO.

Research Participants

Participants were selected using *purposive sampling* based on their involvement in and understanding of the KPI monitoring process. Interviews were conducted with three division heads, namely the Chief Financial Officer (CFO), Chief Customer Officer (CCO), and Chief Operating Officer (COO). In the UAT stage, the three of them (R1 to R3) were joined by one UI/UX expert lecturer-practitioner (R4), giving four respondents. In the system, the CEO acts as the administrator (Admin) who sets KPIs and monitors results, whereas the division heads act as users (User) who enter their divisions' KPI achievements.

Research Instruments

The instruments used consisted of:

1. **Interview guide:** 16 questions in four aspects, namely monitoring method (MP1 to MP4), KPI monitoring (MK1 to MK4), performance reporting (LK1 to LK3), and divisional needs (KD1 to KD5). The interviews were semi-structured so that respondents could add operational context.
2. **Black Box Testing scenarios:** 18 test items across five features, namely login, divisional KPI management, performance input, quarterly evaluation, and the monitoring dashboard.
3. **UAT questionnaire:** 20 statements in four aspects (system functionality, ease of use, performance and reliability, and user satisfaction), with five statements each. A four-point Likert scale was used: Strongly Disagree (1), Disagree (2), Agree (3), and Strongly Agree (4). A scale without a neutral option was chosen so that respondents had to commit to a leaning in their assessment.
4. **Observation:** direct observation of the KPI data entry and monitoring process.

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Data Analysis

1. Qualitative Data Analysis. Interview data were analyzed in three stages, namely data reduction, data display, and conclusion drawing and verification. Responses were grouped by the MP, MK, LK, and KD aspects, merged according to similarity of meaning into problem themes, and then presented in a problem identification table as the basis for formulating system requirements.

2. Quantitative Data Analysis. The Black Box Testing results were analyzed by comparing the actual results with the expected results for each scenario. The UAT results were analyzed using descriptive statistics. The maximum score for each aspect was calculated as the number of statements $\times$ the highest score $\times$ the number of respondents ($5 \times 4 \times 4 = 80$), whereas the minimum attainable score is $5 \times 1 \times 4 = 20$, or 25%, because the Likert scale starts at 1. The percentage was then calculated using the following formula:

$$\text{Percentage (\%)} = (\text{Actual score} \div \text{Maximum score}) \times 100\%$$

Because the lowest attainable percentage is 25% rather than 0%, the category intervals were constructed from the 25% to 100% range divided into four classes corresponding to the number of Likert scale options, namely $(100\% - 25\%) \div 4 = 18.75\%$ per class. The interpretation of percentages follows the categories in Table 2.

Table 2. Interpretation Criteria for UAT Scores

Percentage	Category
25.00% - 43.75%	Very Poor
43.76% - 62.50%	Poor
62.51% - 81.25%	Good
81.26% - 100%	Very Good

Methodological Limitations

This study has the following methodological limitations:

1. There were only four UAT respondents (three division heads and one expert practitioner) in a single business. This number is adequate for an in-depth case study, but the results are interpreted only as an initial indication of acceptance in the Macwish context and not as a statistical generalization. One respondent (R4) is a UI/UX lecturer-practitioner rather than a daily operational user; the UAT scores therefore reflect only the assessments of these four respondents and do not prove that the system is suitable for all users or all operational conditions.
2. Testing was conducted in Macwish's internal environment during the initial implementation period and did not include long-term testing.
3. The evaluation was limited to Black Box Testing and UAT. Performance, load, and security testing, as well as usability testing using standard instruments such as the System Usability Scale (SUS), were not conducted; the high UAT scores are therefore not a guarantee of quality in those aspects. Likewise, the 18 Black Box Testing scenarios did not cover testing of role-based access rights, data security, or system resilience, so their success cannot be taken to guarantee the overall security and reliability of the system.
4. Efficiency was not measured quantitatively, for example in terms of report preparation time or the number of errors before and after using the system. The statement that the system is more effective than spreadsheets is based on user perceptions. This study does not claim improved effectiveness as a proven result; what was demonstrated is limited to functional feasibility and initial user acceptance.

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

5. This study did not compare Base44 with other no-code/low-code platforms or with conventional code-based development, and did not integrate the system with external software such as ERP or HR systems.
6. As a no-code platform, Base44 has more limited flexibility than full programming.

RESULTS

Analysis of the Existing System and User Requirements

KPI monitoring at Macwish Indonesia currently uses spreadsheets to record the targets, actuals, weights, and achievements of each indicator. The process begins with the creation of KPIs and calculation formulas, followed by the entry of achievements by each division, the calculation of achievements using formulas, and recapitulation as material for evaluation. Although data are stored centrally in a single document, updating and supervision depend on manual management. As a result, there is a risk of input and formula errors, information is not updated automatically, and data are presented as tables, so performance analysis takes longer.

The interview results from the three division heads are summarized in Table 3, together with the proposed solutions. Two respondents considered the current method less effective or not yet effective, while one respondent rated it as effective at around 75 percent. All three stated that there is no real-time KPI monitoring system and considered a centralized system important or very important. The eight themes in Table 3 were obtained by grouping responses to the 16 interview questions according to similarity of meaning, and the Respondents column indicates the division heads (CFO, CCO, or COO) whose answers formed the basis of each theme, according to the interview notes.

Table 3. Problem Identification and Proposed Solutions

No	Code	Problem Findings	Solution in the Dashboard	Respondents
1	MP1, MP4	Monitoring is not yet continuous because it is carried out through monthly reports or routine meetings.	A dashboard as a structured monitoring medium.	CFO,CCO, COO
2	MP2, MP3	Spreadsheets are impractical: complicated formula setup, visually unappealing, and document links are hard to find.	A no-code web system accessed through a browser with a simple interface.	CFO,CCO, COO
3	MK1, MK2	There is no real-time monitoring; KPI progress is known through meetings or periodic reporting.	The dashboard displays achievements immediately after the data are updated.	CFO,CCO, COO
4	MK3, KD1	A centralized system is considered important so that progress across divisions can be viewed without switching between documents.	A single centralized dashboard for all divisions.	CFO,CCO, COO
5	MK4, KD2	Required information: indicators, targets, achievements, progress, supporting evidence, and evaluation reports.	Division pages, KPI details, KPI achievements, and quarterly evaluation.	CFO,CCO, COO
6	LK1 - LK3	Manual reporting, forgotten activities not recorded, reporting commitment, and non-standardized formats.	A structured performance input form with an input history.	CFO,CCO, COO
7	KD3, KD4	Graphical visualization is preferred; expected features include evidence of achievement and target alerts.	Trend charts, bar charts, filters, and color-coded achievement indicators. Alert and notification features were not included in the testing.	CFO,CCO, COO

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

No	Code	Problem Findings	Solution in the Dashboard	Respondents
8	KD5	The system is expected to be easy for all users to operate.	A simple interface; assessed under the ease-of-use aspect in the UAT.	CFO,CCO, COO

4.2 System Design

The system has two actors. The **Admin** (CEO) has full access to manage user, division, and KPI data, set targets, and monitor the performance of all divisions. The **User** (CFO, CMO, COO, and CCO) has limited access to enter and view their division's KPI achievements, monitor progress against targets, and view performance visualizations. The functional requirements consist of 12 functions (F-001 to F-012) covering login, data management, achievement entry, dashboard monitoring, and logout, as illustrated in the use case diagram (Figure 1). The non-functional requirements include usability, accessibility (accessible through a browser on both desktop and smartphone), reliability, and user satisfaction.

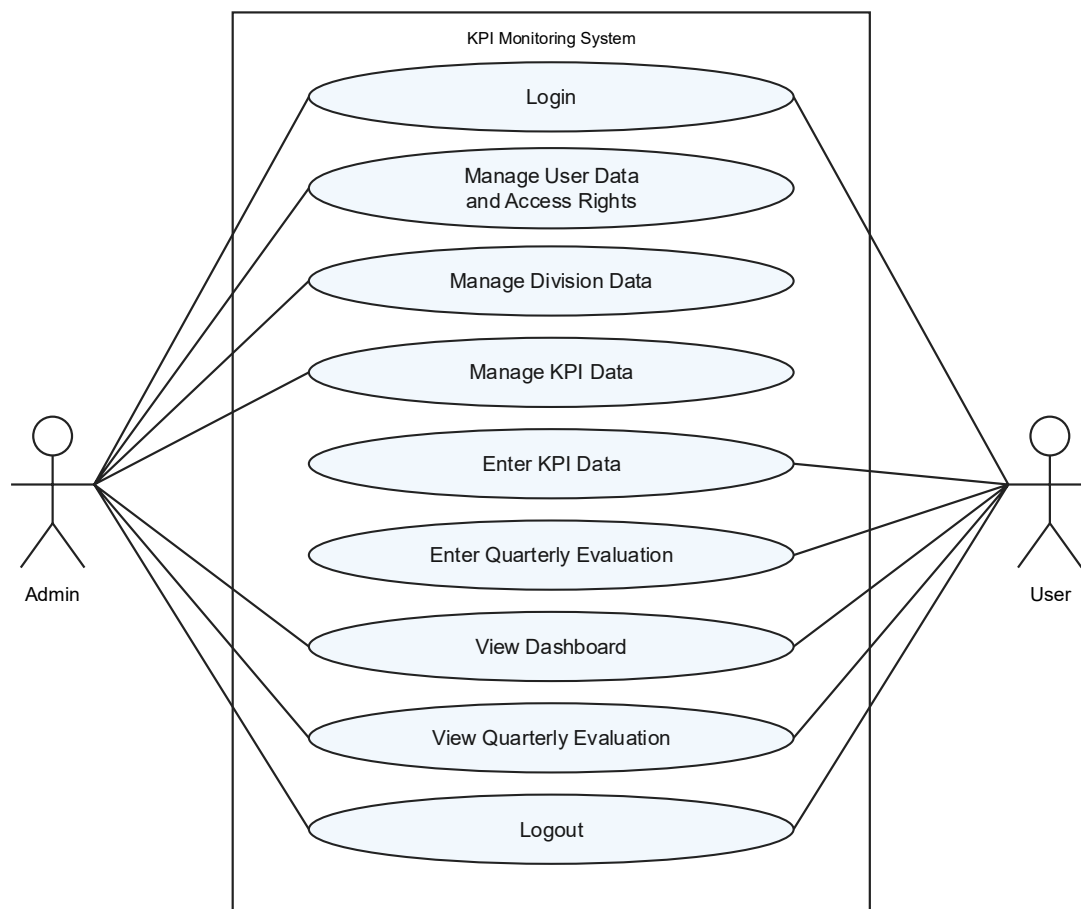


Figure 1 Use Case Diagram of the KPI Dashboard

Data are managed on Base44 in the form of entities. There are six mutually supporting entities, as shown in Table 4.

Table 4. Data Model of the KPI Dashboard

No	Entity	Description
1	User	User accounts and role-based access rights.
2	Division	Division data that are the object of monitoring and evaluation.
3	KPI	Performance indicators with their targets, weights, units, and periods.
4	PerformanceEntry	Periodically entered KPI achievements with supporting evidence.
5	QuarterlyEvaluation	Performance evaluation results of each division for each quarter.
6	KPIEvaluationNote	Evaluation notes, recommendations, and follow-up actions for KPI achievements.

4.3 System Implementation

The dashboard was built on Base44 by entering the requirements resulting from the design stage as text instructions. The system consists of a login page, a main dashboard, a division page, KPI and division management, performance input, and quarterly evaluation.

4.3.1 Login Page. The login page restricts access to registered users only. If the username or password is correct, the user is directed to the main dashboard. If it is incorrect, the system displays an error notification (Figure 2).

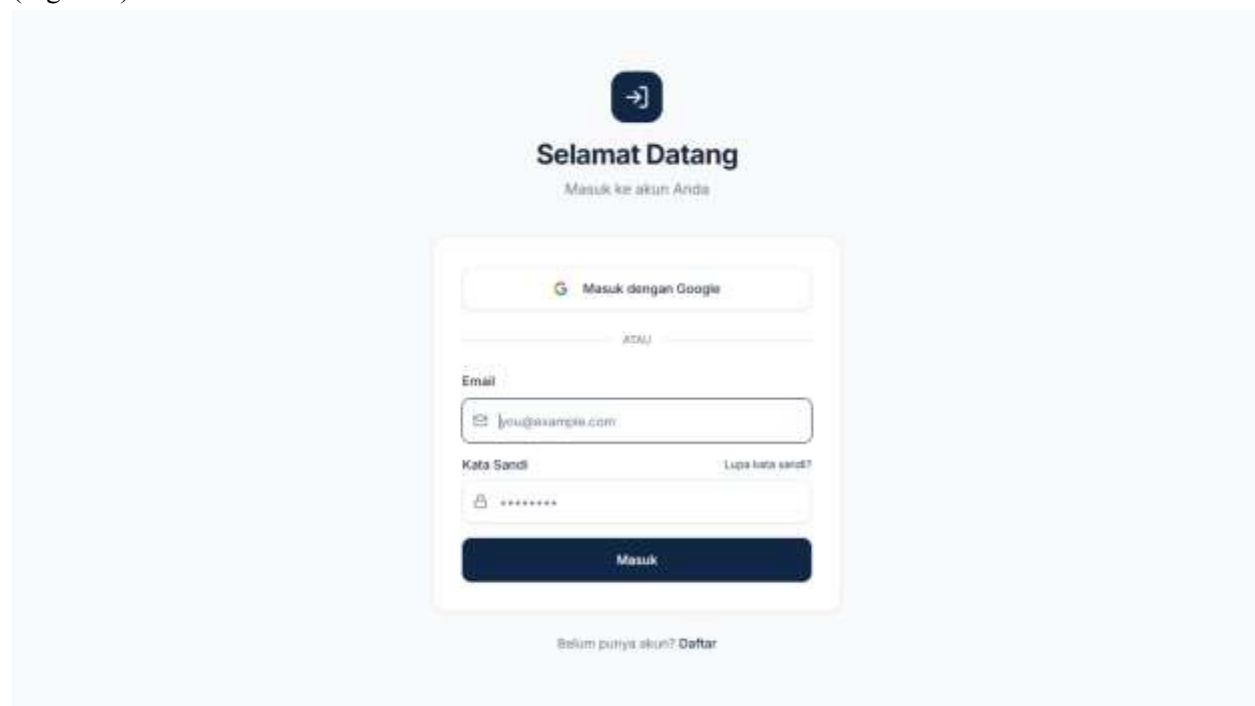


Figure 2. Login Page

4.3.2 Main Dashboard. The main dashboard displays a summary of the number of active KPIs and the target achievement level of all divisions (Figure 3). The monthly KPI trend chart shows the achievement percentages of the four divisions over a full year, complete with a target line and a division filter. The divisional performance bar chart compares achievements per quarter and can be filtered by quarter. Clicking a bar displays the monthly breakdown of KPI targets and achievements, with color indicators for the achievement level. The divisional ranking is sorted by the average KPI achievement in the selected month.

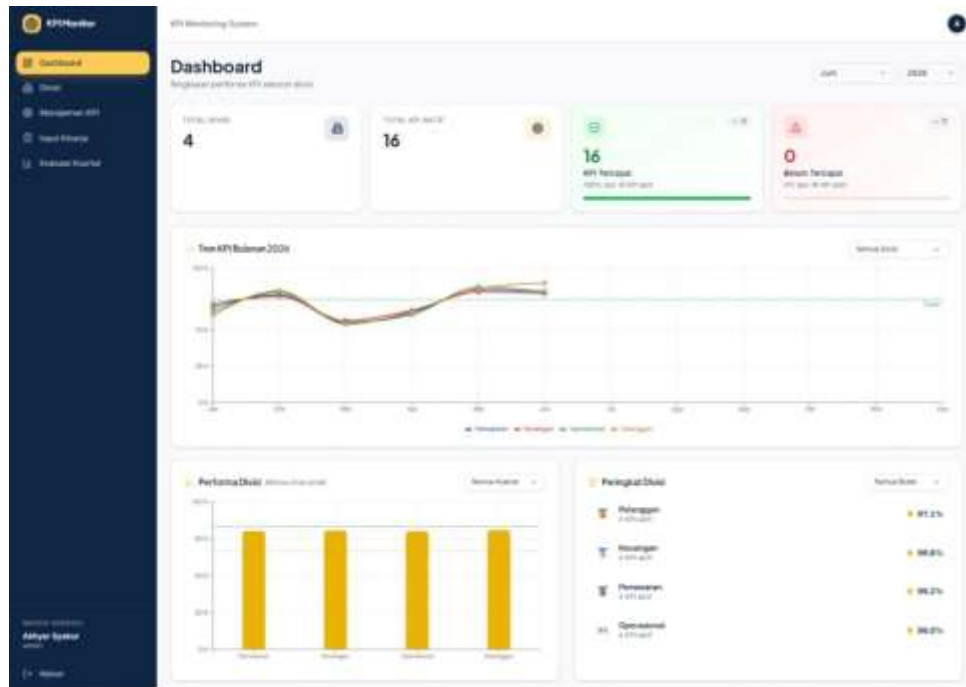


Figure 3. Main Dashboard and Divisional Performance Visualization

4.3.3 Division Page. Each division is displayed as a card containing the division name, KPI achievement level, number of active KPIs, and performance status. The KPI details include the name, description, target, type, maximum achievement percentage, weight, and status. This page also displays the performance input history and the achievement percentage chart (Figure 4).

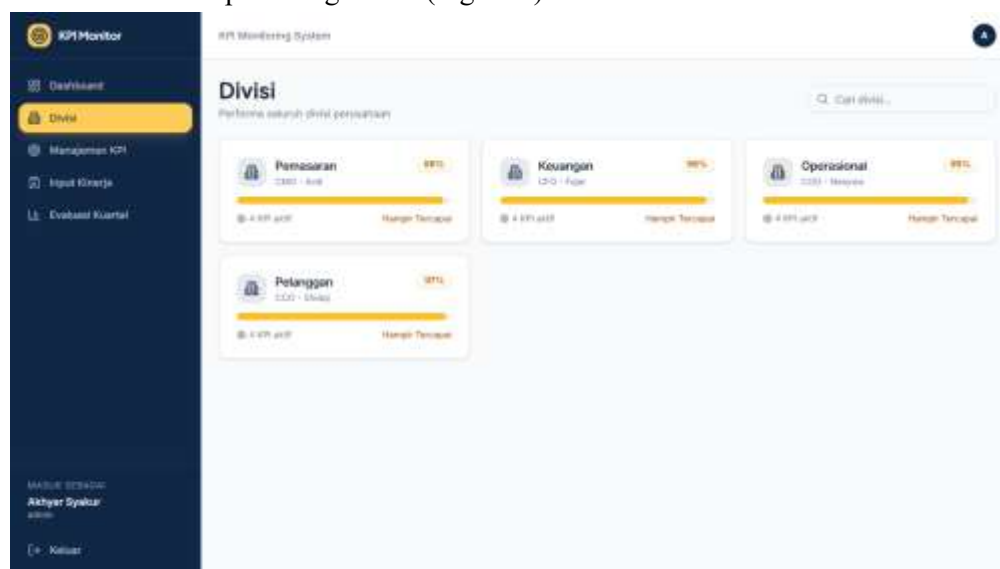


Figure 4. Division Page and KPI Details

4.3.4 KPI and Division Management. The Admin can add, edit, and delete KPIs and divisions (Figure 5). Adding a KPI requires a name, category, target, weight, and calculation method. Data deletion is preceded

* Corresponding author



by a confirmation dialog to reduce the risk of accidental deletion.

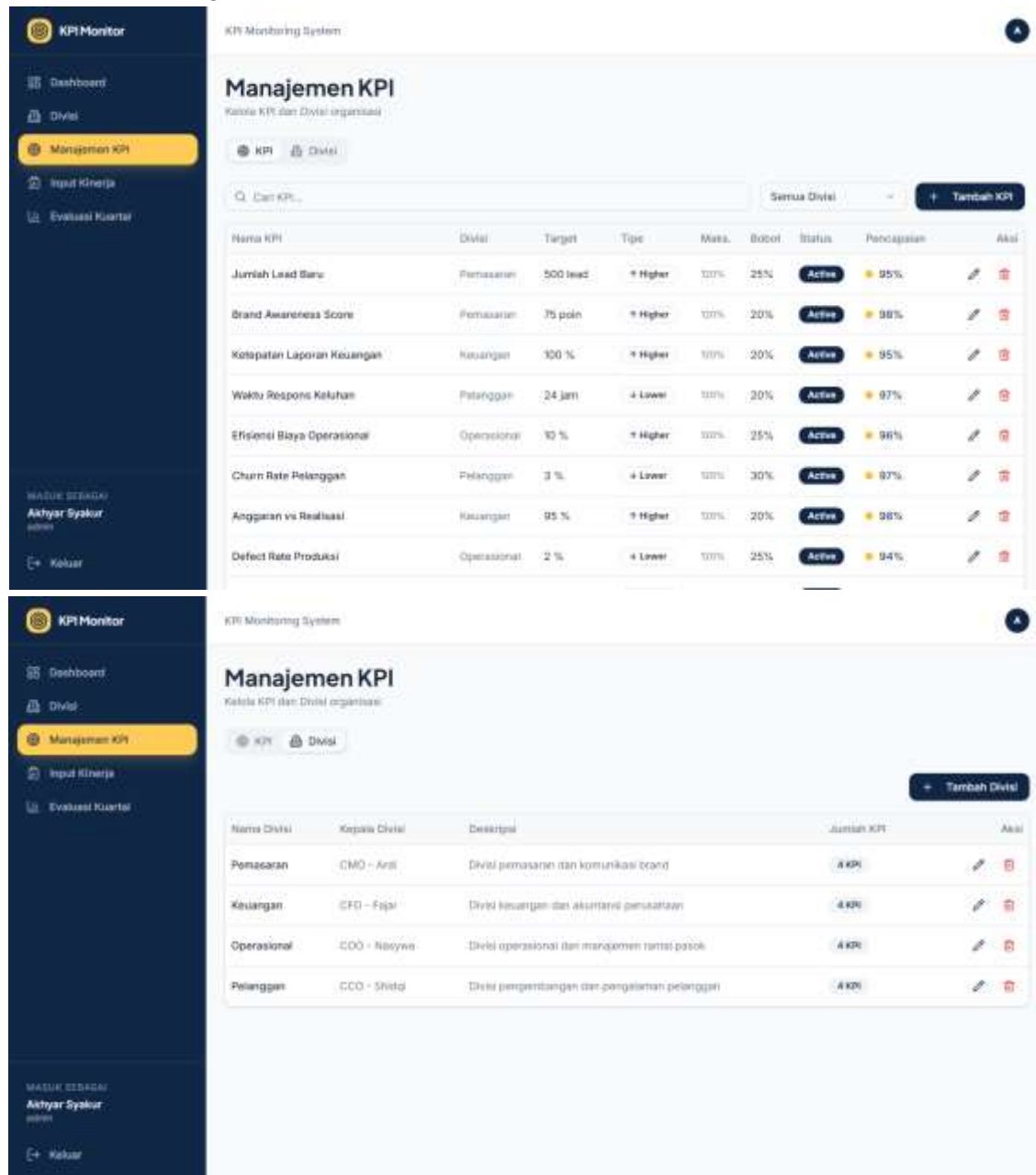
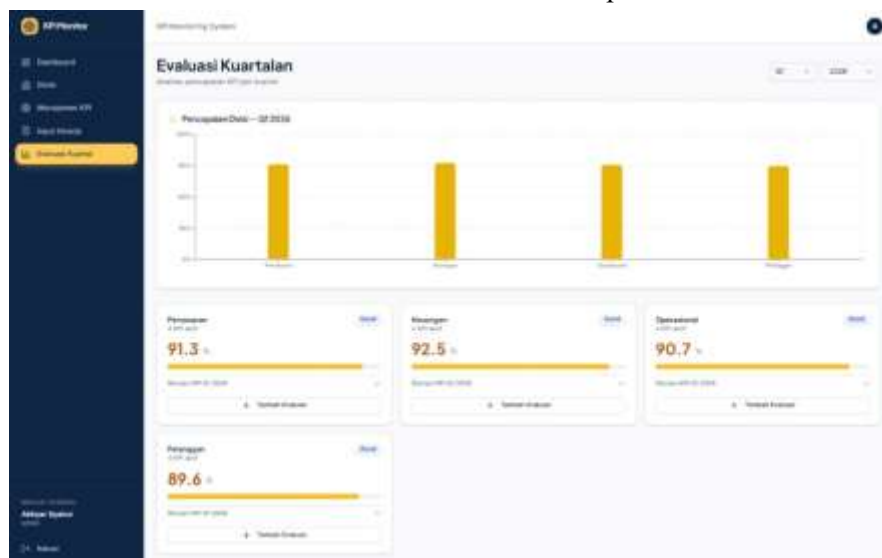


Figure 5. KPI Management Page

4.3.5 Performance Input. This page is used to enter KPI actuals for each period (Figure 6). All information on the dashboard originates from data entered through this page, and every record is automatically saved in the input history.

Figure 6. Performance Input Page

4.3.6 Quarterly Evaluation. This page displays the evaluation results of each division based on KPI achievements over one quarter and allows users to record strengths, weaknesses, and recommendations (Figure 7). These notes serve as evaluation documentation and as input for decisions in the following period.



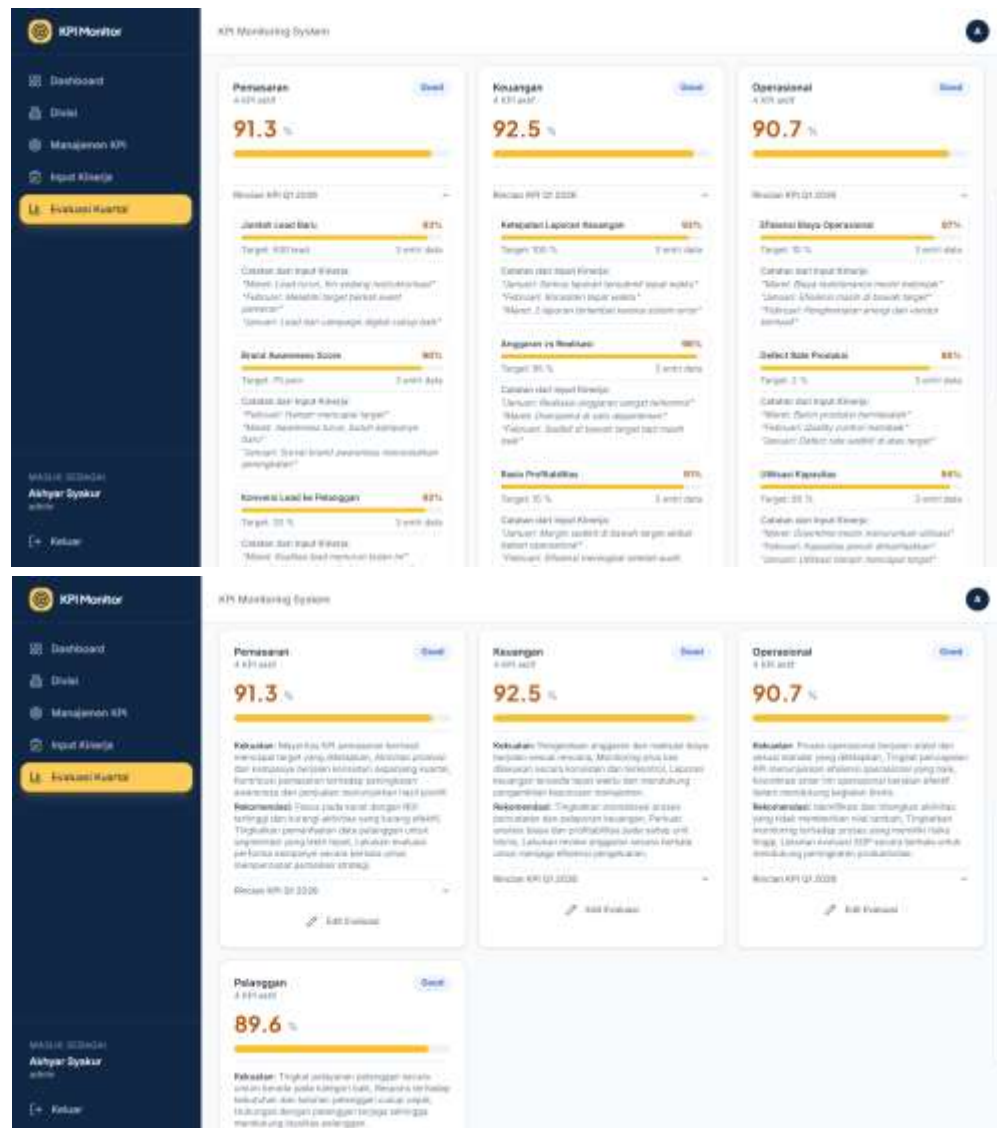


Figure 7. Quarterly Evaluation Page

4.3.7 KPI Achievement Calculation Mechanism. The final division score is calculated in three steps: calculating the achievement of each KPI against its target, multiplying it by the KPI weight, and then summing the weighted scores of all active KPIs of the division. Indicator direction is divided into two types. For indicators where a higher value is better (higher is better), such as the number of portions sold, achievement is calculated with the following formula.

$$\text{KPI Achievement (\%)} = \min [(\text{Actual} \div \text{Target}) \times 100\% ; \text{Maximum achievement}]$$

For indicators where a lower value is better (lower is better), such as service time or the number of complaints, the ratio is inverted so that an actual value smaller than the target yields a higher achievement.

$$\text{KPI Achievement (\%)} = \min [(\text{Target} \div \text{Actual}) \times 100\% ; \text{Maximum achievement}]$$

The maximum achievement (the maximum achievement percentage in the KPI details) caps the achievement value so that an actual value far exceeding the target does not dominate the division score.

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

The score of each KPI and the final division score are calculated as follows.

$$\text{KPI Score} = \text{KPI Achievement} \times \text{KPI Weight}$$

$$\text{Final division score (\%)} = \Sigma (\text{KPI Achievement} \times \text{KPI Weight}) \div \Sigma \text{KPI Weight}$$

Dividing by the sum of weights keeps the division score on a scale of 0% to 100% even if the total weight of active KPIs is not exactly 100%. This final score is used to rank the divisions in the selected month (Section 4.3.2).

As an illustration with hypothetical figures (not Macwish data), a division has three KPIs in one month. The first KPI (number of portions sold, higher is better) has a target of 400 portions, an actual of 360 portions, and a weight of 40%, so its achievement is $360 \div 400 \times 100\% = 90\%$ and its score is $0.40 \times 90\% = 36.0\%$. The second KPI (average service time, lower is better) has a target of 5 minutes, an actual of 6.25 minutes, and a weight of 35%, so its achievement is $5 \div 6.25 \times 100\% = 80\%$ and its score is $0.35 \times 80\% = 28.0\%$. The third KPI (customer satisfaction score, higher is better) has a target of 4.0, an actual of 4.4, and a weight of 25%. Its ratio is 110%, but it is capped at the maximum achievement of 100%, so its score is $0.25 \times 100\% = 25.0\%$. The sum of scores is $36.0\% + 28.0\% + 25.0\% = 89.0\%$ with a total weight of 100%, so the final division score is $89.0\% \div 100\% = 89.0\%$. Without the cap on maximum achievement, this value would be 91.5%, so the capping rule directly affects the ranking results.

It should be noted that the Black Box Testing scenarios in Table 5 test the display of achievement visualizations and filters, but do not explicitly verify the accuracy of the calculated figures against manual calculations.

4.4 Black Box Testing Results

Black Box Testing was conducted on five main features with 18 test items. The actual results were compared with the expected results, and all items passed (Table 5).

Table 5
Summary of Black Box Testing Results

No	Feature	Test Items	Scenario	Status
1	System login	3	Correct credentials; incorrect credentials; empty fields	All passed
2	Divisional KPI management	4	Add, edit, and delete KPIs; save incomplete data	All passed
3	Performance input	4	Enter complete data; edit data; display data; save incomplete data	All passed
4	Quarterly evaluation	3	Add, edit, and display evaluation results	All passed
5	Monitoring dashboard	4	Dashboard access; achievement visualization; division filter; period filter	All passed
	Total	18		18 passed

These results show that the main functions operate in accordance with the functional requirements, including authentication, data management (create, read, update, delete), validation of incomplete data, and dashboard filters. The testing covered only the designed scenarios; system behavior under data volumes or usage patterns outside those scenarios has therefore not been tested. All test items focused on functional correctness. Testing of role-based access rights (Admin and User), data security (for example, password protection and prevention of unauthorized access), and system resilience (for example, user load and data recovery) was not included in the 18 scenarios. The success of the 18 scenarios therefore shows only that the tested functions operate according to the scenarios and cannot be taken to guarantee the overall security

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

and reliability of the system, even though the system manages divisional performance data.

4.5 User Acceptance Testing (UAT) Results

Before completing the questionnaire, respondents tried the main dashboard, KPI management, performance input, division page, and quarterly evaluation features. The score for each statement is presented in Table 6, and the recapitulation per aspect in Table 7.

Table 6. UAT Scores per Statement

Aspect	No	Statement	R1	R2	R3	R4	Total
System Functionality	1	The division page displays all of the division's KPIs in accordance with their targets	4	4	4	3	15
	2	All pages and features can be used properly	4	4	4	3	15
	3	Monthly and annual KPI progress is displayed clearly	4	4	4	4	16
	4	KPI information meets monitoring needs	3	4	4	4	15
Ease of Use	5	The system helps evaluate KPI achievement	4	4	4	4	16
	6	The dashboard is easy to understand when first used	4	4	4	4	16
	7	Navigation between menus is clear	4	4	4	4	16
	8	The dashboard display is neat and not confusing	4	4	4	4	16
	9	KPI information is easy to find	4	4	4	4	16
	10	Easy to use for non-technical users	3	4	4	3	14
Performance and Reliability	11	The KPI dashboard can be accessed smoothly without technical disruption	4	4	4	4	16
	12	The KPI data displayed are consistent and trustworthy	4	4	4	3	15
	13	The system does not experience errors or failures during use	4	4	4	4	16
	14	Changes or updates to KPI data are displayed correctly	4	4	4	4	16
	15	The dashboard can be relied on as a tool for evaluating divisional performance	4	4	4	4	16
User Satisfaction	16	Users are satisfied with using the dashboard	4	4	4	4	16
	17	The dashboard helps in understanding performance conditions	4	4	4	4	16
	18	The dashboard makes it easier to monitor KPI targets	4	4	4	3	15
	19	The features and display meet needs	4	4	4	3	15
	20	Willing to use the dashboard in the future	3	4	4	4	15

Table 7. Recapitulation of UAT Results per Aspect

Aspect	Actual Score	Maximum Score	Percentage	Category
System Functionality	77	80	96.25%	Very Good
Ease of Use	78	80	97.50%	Very Good
Performance and Reliability	79	80	98.75%	Very Good
User Satisfaction	77	80	96.25%	Very Good
Overall	311	320	97.19%	Very Good

All aspects fall into the "Very Good" category (81.26% to 100%), with an overall mean of 97.19% (311 out

* Corresponding author



of 320). The performance and reliability aspect obtained the highest percentage (98.75%), followed by ease of use (97.50%), and then system functionality and user satisfaction (96.25% each). The item with the lowest score was "Easy to use for non-technical users" (14 out of 16, or 87.50%), which still falls into the "Very Good" category. Seven items (numbers 1, 2, 4, 12, 18, 19, and 20) scored 15 out of 16, twelve items (numbers 3, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16, and 17) scored 16, and one item (number 10) scored 14, giving a total score of $7 \times 15 + 12 \times 16 + 1 \times 14 = 311$. The expert practitioner (R4) gave a score of 3 on six items, whereas the three division heads gave a score of 3 on only three items (R1) or on none (R2 and R3).

The score of 97.19% reflects the assessments of four respondents, namely three division heads and one UI/UX lecturer-practitioner (R4) who is not a daily operational user of the system. This figure cannot be read as proof that the system is suitable for all users or all operational conditions, and it does not replace testing in real, long-term use.

DISCUSSION

The results show that the no-code KPI dashboard built on Base44 is functionally feasible and initially acceptable to respondents as a medium for divisional KPI monitoring at Macwish Indonesia. The system centralizes KPI data, calculates and displays achievements in visualizations, and provides quarterly evaluation within a single platform. All Black Box Testing items passed, and the UAT scores fall into the "Very Good" category in all four aspects. These results are consistent with findings that dashboards serve as monitoring tools in performance management (Arputharaj et al., 2024) and that low-code platforms can lower the technical barriers to application development (Ajimati et al., 2025; Hagel et al., 2024). However, this study did not measure whether the system speeds up work, reduces errors, or improves monitoring effectiveness compared with spreadsheets; the findings that can be stated are therefore limited to functional feasibility and initial user acceptance, not improved effectiveness.

In terms of acceptance, the high ease-of-use score (97.50%) indicates a positive assessment by respondents of the dashboard's ease of use. This aspect is conceptually related to the TAM framework, which positions ease of use and usefulness as the main determinants of acceptance (Mogaji et al., 2024). However, the UAT instrument in this study measures only four system evaluation aspects and does not measure the full set of TAM constructs, so the TAM serves as a conceptual foundation and these results are not an empirical test of the TAM. The interview findings also show that user needs lie not only in recording but also in fast access, clear target and achievement information, and graphical visualization. However, the item "easy to use for non-technical users" in fact obtained the lowest score (87.50%). This suggests that interface simplification can still be improved, especially on the management and evaluation pages, which contain many input fields. This value should be interpreted with caution because it comes from four respondents.

Compared with previous studies whose testing was limited to system functions (Aryanto & Susila, 2024) or that used code-based development (Fakhruddin, 2025), this study produced a dashboard that has been implemented and tested with a real-world partner without manual programming. Its practical contribution is a centralized KPI monitoring tool for the CEO and division heads, as well as a replicable example for small and medium-sized F&B businesses with limited IT resources. Its scientific contribution is contextual, in the form of documentation of the application of no-code to a divisional KPI dashboard, together with the findings from its implementation and evaluation described in the following paragraph. At the international level, KPI dashboards have also been reported to help management make decisions faster (Nunes et al., 2024), and user involvement from requirement elicitation through usability testing has become a consistent practice (Bennett & Müller Loose, 2024). This is in line with the interview, implementation, and UAT flow of the present study. The lowest score on the item concerning ease of use for non-technical users should be read together with the note by Ajimati et al. (2025) that the adoption of low-code/no-code still entails challenges alongside its benefits, and together with the research direction that uses LLMs to lower the technical expertise barrier for developers (Hagel et al., 2024).

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

This implementation and evaluation yielded four findings that go beyond a description of the application's features. First, interview results could be translated directly into text instructions on Base44 that produced six entities and 12 functional requirements, so that traceability from requirement themes (Table 3) to system features (Section 4.3) could be maintained without manual programming. This finding suggests that, in text-description-based approaches, the quality of the outcome depends on the care taken in specifying requirements. Second, the rules that determine the correctness of results, namely indicator direction, the cap on maximum achievement, and weighting (Section 4.3.7), still have to be formulated and verified explicitly. The ease of building an interface does not remove the need to validate calculation logic, and in this study such validation was not demonstrated through dedicated test scenarios. Third, ease of building an application does not automatically mean ease of using it. The item on ease of use for non-technical users obtained the lowest score (87.50%), and the UI/UX practitioner (R4) gave a score of 3 on six items, more than the three division heads, who gave a score of 3 on three items (R1) or on none (R2 and R3). This suggests that assessors with an interface design background are more critical of the interface quality produced by the platform, so evaluations of no-code systems should involve assessors with different perspectives rather than end users alone. Fourth, the success of the 18 Black Box Testing scenarios and the high UAT scores show that a no-code platform can produce a system that functions according to requirements within the tested scope, but neither evaluation answers the core question for small businesses, namely whether monitoring becomes faster and more accurate than with spreadsheets. This question needs to be answered in future research by measuring time and errors before and after implementation.

This study has several limitations. First, the UAT involved only four respondents in a single business, so the results constitute preliminary evidence and cannot be generalized. Second, the advantage over spreadsheets and the development speed of no-code were not measured quantitatively (time, errors, or cost) and were not compared with other platforms or code-based development, so these claims are perceptual. Third, the evaluation was limited to Black Box Testing and UAT during the initial implementation period; system performance under high load, data security, role-based access rights, system resilience, and long-term use have not been tested. Fourth, the system does not yet include automated analysis, decision recommendations, or integration with other systems.

Future research agenda includes: (1) evaluation in long-term use with more respondents; (2) measurement of efficiency before and after implementation, along with additional testing such as SUS, performance, and security testing (including role-based access rights and system resilience), as well as verification of the accuracy of KPI achievement calculations against manual calculations; (3) development of trend analysis, root cause analysis, and automated recommendations; (4) integration with ERP and HRIS systems, including individual KPIs; and (5) comparative studies across no-code/low-code platforms.

CONCLUSION

This study successfully developed a no-code KPI dashboard using Base44 for divisional performance monitoring at Macwish Indonesia using the Waterfall model. The dashboard provides login, division and KPI management, performance input, monitoring dashboard, and quarterly evaluation features that centralize KPI data and present them in visualizations.

All 18 Black Box Testing items passed in the functional scenarios tested, which did not cover access rights, data security, or system resilience. The UAT results from three division heads and one expert practitioner show scores of 96.25% for system functionality, 97.50% for ease of use, 98.75% for performance and reliability, and 96.25% for user satisfaction, with an overall mean of 97.19% ("Very Good" category). These results indicate that a no-code approach has the potential to serve as an alternative for developing performance monitoring systems for non-technical users in similar contexts. This study has so far demonstrated only functional feasibility and initial user acceptance based on the assessments of four respondents, and has not demonstrated improved monitoring effectiveness compared with spreadsheets.

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

The implementation findings show that the care taken in specifying requirements and KPI calculation rules determines the correctness of results, and that ease of building an application does not automatically mean ease of use for non-technical users. However, the small number of respondents, the absence of quantitative efficiency measurement, and the absence of comparison with other platforms mean that these findings cannot yet be generalized.

Future studies are recommended to conduct long-term evaluations with more respondents, measure efficiency quantitatively, add trend analysis and automated recommendation features, and integrate the dashboard with ERP and HRIS systems to support KPIs at both the divisional and individual levels.

ACKNOWLEDGMENTS

The authors would like to thank Sekolah Tinggi Teknologi Terpadu Nurul Fikri for its academic support throughout this study. The authors are also grateful to Macwish Indonesia and its division heads, who served as partners, took the time to be interviewed, and provided data, as well as to the UI/UX expert practitioner who participated in the system testing.

REFERENCES

- Ajimati, M. O., Carroll, N., & Maher, M. (2025). Adoption of low-code and no-code development: A systematic literature review and future research agenda. *Journal of Systems and Software*, 222, 112300. <https://doi.org/10.1016/j.jss.2024.112300>
- Arputharaj, J. V., Yakub, M. E., Haruna, A. A., & Senthil Kumar, A. (2024). Review and design of integrated dashboard model for performance measurements. In H. Zen, N. M. Dasari, Y. M. Latha, & S. S. Rao (Eds.), *Soft computing and signal processing: ICSCSP 2023* (Lecture Notes in Networks and Systems, Vol. 840, pp. 1–10). Springer. https://doi.org/10.1007/978-981-99-8451-0_1
- Aryanto, M., & Susila, A. (2024). Perancangan dashboard visualisasi data key performance indicator (KPI) menggunakan Looker Data Studio (studi kasus PT Sicepat Ekspres) [Design of a key performance indicator (KPI) data visualization dashboard using Looker Data Studio (case study of PT Sicepat Ekspres)]. *OKTAL: Jurnal Ilmu Komputer dan Science*, 3(1), 204–224.
- Bennett, A. W., & Müller Loose, S. (2024). User-centered development of an online dashboard tool for economic sustainability for small and medium enterprises. *Sustainability*, 16(2), 557. <https://doi.org/10.3390/su16020557>
- Domański, R., Wojciechowski, H., Lewandowicz, J., & Hadaś, Ł. (2023). Digitalization of management processes in small and medium-sized enterprises—An overview of low-code and no-code platforms. *Applied Sciences*, 13(24), 13078. <https://doi.org/10.3390/app132413078>
- Fakhrudin, M. A. (2025). Implementasi metode key performance indicator pada aplikasi penilaian karyawan berbasis web di PT Sentra Vidya Utama [Implementation of the key performance indicator method in a web-based employee appraisal application at PT Sentra Vidya Utama] [Undergraduate thesis, Universitas Dinamika]. Universitas Dinamika Repository. <https://repository.dinamika.ac.id/id/eprint/8246/>
- Hagel, N., Hili, N., & Schwab, D. (2024). Turning low-code development platforms into true no-code with LLMs. In *MODELS Companion '24: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*. ACM. <https://doi.org/10.1145/3652620.3688334>
- <https://doi.org/10.23969/jp.v9i04.19906> Husnayayin, A., Gustina, Z., & Dewi, D. E. C. (2024). Karakteristik dan langkah-langkah metode penelitian research and development (Borg & Gall)

- dalam pendidikan [Characteristics and steps of the research and development (Borg & Gall) method in education]. *Pendas: Jurnal Ilmiah Pendidikan Dasar*, 9(4), 490–501. <https://doi.org/10.23969/jp.v9i04.19906>
- Kristofher, Somantri, A., & Sukenda, A. (2022). Perancangan dashboard monitoring kinerja pegawai (studi kasus: Perusahaan Dagang XXX) [Design of an employee performance monitoring dashboard (case study: Trading Company XXX)]. *Pasinformatik*, 1(2), 9–16.
- Marino-Romero, J. A., Palos-Sánchez, P. R., & Velicia-Martín, F. A. (2024). Evolution of digital transformation in SMEs management through a bibliometric analysis. *Technological Forecasting and Social Change*, 199, 123014. <https://doi.org/10.1016/j.techfore.2023.123014>
- Mogaji, E., Viglia, G., Srivastava, P., & Dwivedi, Y. (2024). Is it the end of the technology acceptance model in the era of generative artificial intelligence? *International Journal of Contemporary Hospitality Management*, 36(10), 3324–3339. <https://doi.org/10.1108/IJCHM-08-2023-1271>
- Nunes, F., Alexandre, E., & Gaspar, P. D. (2024). Implementing key performance indicators and designing dashboard solutions in an automotive components company: A case study. *Administrative Sciences*, 14(8), 175. <https://doi.org/10.3390/admsci14080175>
- Saravanos, A. (2025). A brief history of the waterfall model: Past, present, and future. *arXiv*. <https://arxiv.org/abs/2510.03894>
- Silva, A., Maldonado, I., da Silva, M., & Cepeda, C. (2025). Sustainability balanced scorecard: Systematic literature review. *Journal of Risk and Financial Management*, 18(6), 324. <https://doi.org/10.3390/jrfm18060324>
- Yoewono, G. F., Sutrisno, T., & Lewenusa, I. (2024). Perancangan dashboard interaktif untuk pemantauan kedisiplinan karyawan di PT XYZ [Design of an interactive dashboard for monitoring employee discipline at PT XYZ]. *Costing: Journal of Economic, Business and Accounting*, 7(6), 6164–6175.