DESIGN AND CONSTRUCTION OF A FREELANCE MOBILE APPLICATION BASED ON THE KANO MODEL AND THE RAPID APPLICATION DEVELOPMENT (RAD) METHOD

Badrus Zaman[1)], Brilian Febrianie[2)], Endah Purwanti[1,3)] dan Octavia Putri Yudhistira[4)]
[1, 2, 3,4)] Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Airlangga
Surabaya, Indonesia
[1, 3)] Center For Information Systems Engineering, Universitas Airlangga, Surabaya, Indonesia
e-mail: badruszaman@fst.unair.ac.id [1)]

ABSTRAK

Pekerja lepas di Indonesia menghadapi tantangan dalam menjangkau klien secara luas akibat keterbatasan platform yang ada. Platform freelance yang tersedia masih terbatas pada jenis pekerjaan tertentu dan belum mengintegrasikan pendekatan berbasis kebutuhan pengguna secara menyeluruh. Penelitian ini bertujuan merancang dan mengembangkan aplikasi mobile penghubung antara penyedia dan pencari jasa kerja lepas dengan menerapkan Model Kano dan metode Rapid Application Development (RAD). Kombinasi kedua metode ini dalam konteks platform freelance berbasis mobile merupakan pendekatan yang belum banyak diterapkan secara terpadu di Indonesia. Model Kano digunakan untuk memetakan prioritas kebutuhan pengguna, sedangkan RAD digunakan untuk mempercepat pengembangan sistem melalui iterasi yang melibatkan pengguna secara langsung. Penelitian ini melibatkan 52 responden dalam tiga tahap: Requirement Engineering, analisis Model Kano, dan pengembangan sistem RAD sebanyak dua siklus iterasi. Hasil analisis Model Kano terhadap 8 atribut menghasilkan 5 fitur Must-Be, 1 fitur One-Dimensional, 1 fitur Attractive, dan 1 fitur Indifferent yang tidak diprioritaskan, sehingga diperoleh 7 fitur prioritas pengembangan. Pengujian User Acceptance Test (UAT) menghasilkan nilai rata-rata 95,9% dengan kategori Sangat Baik. Penelitian ini berkontribusi pada kerangka metodologis integratif Kano-RAD sebagai acuan pengembangan aplikasi berbasis kebutuhan pengguna pada domain platform jasa digital.

***Kata Kunci**: aplikasi mobile pekerja lepas, analisis kebutuhan, user acceptance test (UAT), model kano, rapid application development (RAD),*

ABSTRACT

Freelance workers in Indonesia face challenges in reaching clients on a wide scale due to the limitations of existing platforms. Available freelance platforms remain restricted to certain types of work and have not yet fully integrated a user needs based approach. This study aims to design and develop a mobile application that connects providers and seekers of freelance services by applying the Kano Model and the Rapid Application Development method, also known as RAD. The combination of these two methods within the context of a mobile based freelance platform represents an approach that has not been widely applied in an integrated manner in Indonesia. The Kano Model is used to map user need priorities, while RAD is used to accelerate system development through iterations that directly involve users. This study involved 52 respondents across three stages, namely Requirement Engineering, Kano Model analysis, and RAD system development consisting of two iteration cycles. The Kano Model analysis of 8 attributes resulted in 5 Must Be features, 1 One Dimensional feature, 1 Attractive feature, and 1 Indifferent feature that was not prioritized, yielding 7 priority features for development. User Acceptance Testing, also known as UAT, produced an average score of 95.9%, categorized as Very Good. This study contributes an integrative Kano RAD methodological framework as a reference for user needs based application development within the digital service platform domain.

***Keywords**: freelancers mobile application, kano model, rapid application development (RAD), requirement analysis, user acceptance test (UAT)*

I. INTRODUCTION

According to data from the Central Bureau of Statistics (BPS), Indonesia's labor force is still dominated by informal workers or freelancers. The proportion of informal workers in Indonesia from 2019 to 2022 consistently exceeded 55%, reaching 59.31% in 2022 out of a total labor force of 144.01 million people, or approximately 85.41 million individuals (BPS, 2022). This dominance indicates that the majority of the workforce in Indonesia relies on freelance or informal employment.

However, freelancers face various challenges, including difficulty in reaching clients, income uncertainty, lack of transparency in project pricing, and difficulties in building trust due to the prevalence of misinformation (Wijaya et al., 2022). These conditions make it difficult for many freelancers to earn a sustainable income due to limited access to job markets and the lack of platforms that effectively facilitate connections between workers and clients.

As a solution, it is necessary to develop an application that functions as a platform connecting freelancers and service seekers. The platform is designed to create a transparent, efficient, and user-oriented system, enabling freelancers to access broader markets, find clients that match their criteria, and increase their income opportunities. Reuschke et al. (2017) state that the use of mobile devices and social media can enhance freelancers' productivity and efficiency in self-promotion and network expansion. In Malaysia, Hanip et al. (2022) developed the MyFreelancer application as an online marketing platform for freelancers, as social media alone is not sufficiently effective in reaching broader job markets. In Indonesia, Sribulancer.com serves as an example of a platform connecting clients with creative professionals such as graphic designers, photographers, and content writers. However, the platform is still limited in terms of job variety as it primarily focuses on the creative sector.

Previous studies have also shown that the need for a local mobile-based freelance platform is increasingly urgent. Aristi and Pratama (2021) found that digital freelance marketplaces are more effective than social media in connecting freelancers with clients in the professional services sector. Meanwhile, Bachmid et al. (2022) developed a web-based part-time job search application for SMEs, but this approach is considered limited in terms of user experience compared to native mobile applications. Furthermore, Nurdiansyah et al. (2024) emphasized that mobile-based platforms developed by local developers are essential to address data security risks and payment system barriers often encountered in foreign platforms.

Unlike existing freelance marketplace platforms that primarily provide standard service listing and transaction functionalities, the proposed application was developed through a user-centered requirement engineering process combined with Kano Model analysis. This approach ensures that the implemented features are directly aligned with user expectations and priorities. The study identified several user-preferred features, including freelance service recommendations, worker selection, rating and review mechanisms, reporting functionality, and order history management. These features were not determined solely by developers but were derived from user requirements and subsequently prioritized based on their influence on user satisfaction. In particular, the freelance service recommendation feature was classified as an Attractive attribute, indicating its potential to increase user satisfaction beyond basic platform expectations. Furthermore, the proposed solution is developed as a mobile application, enabling users to access freelance services more conveniently through smartphones, support user mobility, facilitate faster interactions between freelancers and service seekers, and improve accessibility compared to conventional web-based platforms. Therefore, the contribution of this study lies not only in developing a freelance marketplace application but also in applying a user-centered approach through Requirement Engineering and the Kano Model to identify and prioritize features that reflect actual user needs and preferences.

This study develops a mobile freelance marketplace application with features tailored to user needs, such as transparent transaction processes, ease of searching services by category, location-based matching between clients and freelancers, and direct communication channels between users. Additionally, according to JMango360 (2018) cited in Irwinansyah et al. (2020), 90% of mobile device usage time is spent on applications rather than web browsers.

In its development, this study applies the Kano Model to analyze user needs and priorities, as well as the Rapid Application Development (RAD) method to accelerate the design process through iterative development involving users directly. Identifying user needs is a crucial step in information system development to ensure that the system aligns with user expectations (Purwanti & Zaman, 2016). The Kano Model helps classify system attributes based on their impact on user satisfaction (Rif'ah et al., 2020; Zainah et al., 2023). This approach allows developers to focus resources on features that significantly influence user experience while avoiding over-engineering. However, the Kano Model has limitations, as the calculation of coefficients and quadrant analysis can be relatively complex, making it more suitable for a limited number of respondents (Al Anshari et al., 2024).

Meanwhile, RAD method is chosen for its ability to accelerate the development cycle through prototyping, iterative testing, and rapid user feedback (Suwandi et al., 2022; Munawir et al., 2023; Profita et al., 2022). This approach enables developers to produce systems that are flexible, adaptive to changing requirements, and more aligned with user expectations.

The combined application of the Kano Model and RAD method in mobile application development has been proven effective in several studies. The use of the Kano Model in analyzing mobile application service quality helps identify attributes that most influence user

satisfaction, allowing development teams to prioritize features more effectively (Riri et al., 2022; Victoria et al., 2022). Meanwhile, the RAD method, which emphasizes early user involvement, has been shown to produce systems that better meet real user needs and significantly reduce development time (Nugroho et al., 2023; Natalia & Putranto, 2023).

Previous studies have not explicitly integrated Kano Model and RAD method in developing a mobile-based freelance marketplace platform, particularly within the Indonesian context. Most existing studies apply either user needs analysis or agile development methods independently, without combining both within a unified framework that simultaneously addresses feature prioritization and iterative system development.

Therefore, this study offers three main contributions. First, it proposes an integrative Kano-RAD methodological framework as a replicable approach for user-priority-driven mobile application development in the digital service platform domain. Second, it demonstrates the application of user-centered requirement engineering combined with Kano Model analysis to systematically prioritize features based on their impact on user satisfaction. Third, it introduces location-based freelance matching as a key functional feature derived directly from user requirements, facilitating more efficient and geographically relevant connections between freelancers and service seekers.

By integrating the Kano Model and RAD method, this study aims to develop a mobile application that effectively bridges the needs between freelancers and service seekers. It is expected that this platform will increase job opportunities for freelancers, expand market access, and contribute to improving their overall welfare.

II. RESEARCH METHODOLOGY

In general, this study is conducted through three main stages: requirement engineering, feature priority analysis, and architecture and application development.

A. Requirement Engineering

The requirement engineering process consists of four main stages: requirement gathering through interviews and documentation, identification of key requirements to determine important attributes, categorization of attributes and features in more detail, and requirement verification through follow-up interviews. If the verification results do not align with user needs, a re-identification process is carried out to determine more relevant attributes. This process is also used to measure the validity of each attribute based on the validity score criteria, as shown in Table 1 (Hutabri, 2022).

Data collection was conducted through a survey method using questionnaires distributed to two target groups, namely the general public and freelance workers. The sampling method used was purposive sampling, in which respondents were selected based on their relevance to the research context, specifically individuals who have experience as freelance service seekers or providers. Respondents were required to be adults, as adult age was used as the reference criterion to ensure the validity and accuracy of the responses. A total of 52 respondents participated across all stages of the study. The respondent age distribution was as follows: 8 respondents aged 17–20 years, 43 respondents aged 21–24 years representing 82.7% of the total sample, and 1 respondent aged 25–30 years. These two groups, namely general users and freelance workers, were included to capture both the service-seeker and service-provider perspectives.

Table 1. Validity Score Interpretation Criteria

No	Percentage	Description
1.	90% - 100%	Very Valid
2.	80% - 89%	Valid
3.	65% - 79%	Fairly Valid
4.	55% - 64%	Less Valid
5.	≤ - 55%	Not Valid

Instrument validity in this study was assessed using content validity through a verification survey, in which respondents confirmed whether each proposed attribute aligned with their actual needs. An attribute was considered valid if the percentage of affirmative responses met the threshold defined in Table 1. This approach is consistent with the Kano Model methodology, where attribute validation is based on user agreement rather than statistical correlation, given the exploratory nature of the requirement engineering phase.

B. Feature Priority Scale Analysis

The feature priority analysis process in this study utilizes the Kano Model to deeply explore user needs and determine the importance level of each feature to be implemented in system development in accordance with user expectations. In the application development process, the designed features are divided into two categories, namely core features and user preference features.

Core features include essential components that must be present in the system, such as Register, Login, Freelancer/Worker Registration, On-Going Order, Search, Communication, and Cancel/Accept Order. These core features are identified through an analysis of existing freelancer platforms and websites, serving as a fundamental reference in the application design. Meanwhile, user preference

features are additional features proposed based on user expectations and specific needs.

The implementation of the Kano Model in this study consists of four stages, namely:

a. Questionnaire Development

The development of functional and dysfunctional questionnaires is carried out based on the identified user needs. These needs are then categorized into several attributes, which are incorporated into the questionnaire for further analysis.

b. Kano Model Classification Mapping

The classification mapping of questionnaire results is conducted based on the Kano Model, as illustrated in Figure 1. The interpretation of the evaluation table is as follows: if attribute "x" in the functional questionnaire is rated as 1, and in the dysfunctional questionnaire is rated as 2, then the attribute is classified into the Attractive category.

Customer Requirements		Dysfunctional				
		1	2	3	4	5
		I like it	*I expect it*	*I am netral*	*I can tolerate it*	*I dislike it that way*
Functional	1 *I like it*	Q	A	A	A	O
	2 *I expect it*	R	I	I	I	M
	3 *I am netral*	R	I	I	I	M
	4 *I can tolerate it*	R	I	I	I	M
	5 *I dislike it that way*	R	R	R	R	Q

Description:

Q = Questionable A = Attractive

R = Reverse I = Indifirent

O = One dimensional M = Must be

Figure 1. Mapping of Questionnaire Results Classification.

c. Kano Quadrant Mapping

The Kano quadrant mapping process is carried out by calculating the Better coefficient and Worse coefficient for each attribute. The formulas for the Better and Worse coefficients are presented in Equation (1) and Equation (2), respectively. Subsequently, each attribute is mapped into coordinates (X, Y) within the Kano quadrant, as illustrated in Figure 2, where the X-axis represents the results of the dysfunctional questions, and the Y-axis represents the results of the functional questions.

$$\text{Better} = \frac{A+O}{A+O+M+I} \quad (1)$$

$$\text{Worse} = -\frac{M+O}{A+O+M+I} \quad (2)$$

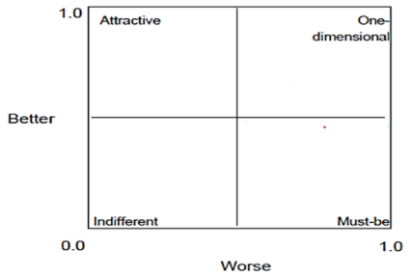

Figure 2. Attribute Mapping in the Kano Quadrant.

d. Evaluation of Kano Classification Results

The evaluation of features based on the Kano quadrant classification is conducted to determine which attributes should be included in the system. Based on the Kano Model, the selected attributes belong to the categories of One-Dimensional (O), Attractive (A), and Must-Be (M), as these categories have a significant impact on user satisfaction and system performance.

C. Architecture and Application Development

The results of the prioritization of key attributes are used as the foundation for developing the system architecture and application using RAD method. This method is implemented through the following five stages:

a. Business Design

To design the business process flow, a use-case diagram is utilized to model the interactions between users or stakeholders and the system.

b. Data Design

Data design is carried out using an Entity Relationship Diagram (ERD) to represent the structure and relationships of the data within the system).

c. Prototyping

At the prototyping stage, a functional prototype is developed and presented to users to obtain feedback. Based on this feedback, adjustments and improvements are made to the prototype to ensure that user requirements are effectively met.

d. Construction

In the construction phase, the application system is developed based on the previously designed models and refined prototypes using Flutter Framework with Dart as the programming language and Firebase Firestore as the cloud-based database. The development environment used is Visual Studio Code, while user interface designs and prototypes were created using Figma and tested on Android-based mobile devices.

e. Cut-over

At the cut-over stage, system testing is conducted using a Likert Scale approach through User Acceptance Testing (UAT). The evaluation is performed by end users to ensure that the application functions properly and meets user expectations. The testing process involves distributing questionnaires to users, and the results are interpreted based on the score criteria shown in Table 2 (Abraham et al., 2021).

Table 2. Score Interpretation Criteria

No.	Percentage	Description
1.	0% - 20%	Very Poor
2.	21% - 40%	Poor
3.	41% - 60%	Fair
4.	61% - 80%	Good
5.	81% - 100%	Very Good

III. RESULT AND DISCUSSION

A. Requirement Engineering

The process of gathering and exploring user requirements was conducted through interviews using open-ended questionnaires distributed to respondents. A total of 52 respondents were asked to complete the questionnaire based on their expectations and needs for the application. Based on the results, several required attributes were identified, as presented in Table 3.

Table 3. Attributes Based on Requirement Analysis.

No.	Feature	Code
1.	Order Description	F1
2.	Order History	F2
3.	Freelance Service Recommendation	F3
4.	User Help	F4
5.	User Guidance	F5
6.	Notification / Warning Message	F6
7.	Rating / Review	F7
8.	Report	F8

B. User Requirement Validity Testing

Validity testing was conducted to assess the reliability of the customer voice results and to validate the proposed attributes to ensure they align with user needs. This process was carried out through a survey to verify each proposed attribute, with the level of validity measured based on the validity score criteria presented in Table 1. The test results indicate an average validity percentage of 98.7%, which falls into the "Very Valid" category. This demonstrates that all proposed attributes are appropriate and feasible for use in this study. The recapitulation of the validity test results is shown in Table 4.

C. Feature Priority Analysis

a. Kano Model Analysis

The Kano Model analysis was conducted by classifying the results obtained from the questionnaire. The responses were categorized into six groups: One-Dimensional (O), Attractive (A), Must-Be (M), Indifferent (I), Reverse (R), and Questionable (Q). The results of this analysis are presented in Table 5.

Table 4. Results of User Requirement Validity Testing.

No.	Code	Total Respondents		Percentage of "Yes"
		Responses "Yes"	Responses "No"	
1.	F1	52	-	100%
2.	F2	49	3	94,2%
3.	F3	52	-	100%
4.	F4	52	-	100%
5.	F5	52	-	100%
6.	F6	50	2	96,1%
7.	F7	52	-	100%
8.	F8	52	-	100%
			Average	98,7%

Table 5. Kano Model Analysis Results.

Code	Attribute	A	O	M	I	R	Q	Total
F1	Order Description	11	11	13	11	0	6	52
F2	Order History	10	15	11	10	0	6	52
F3	Freelance Service Recommendation	15	9	8	14	1	5	52
F4	User Help	5	11	18	13	0	5	52
F5	User Guidance	8	8	17	13	1	5	52
F6	Notification /Warning Message	10	6	12	16	3	5	52
F7	Rating / Review	5	10	14	13	4	6	52
F8	Report	5	7	21	11	3	5	52

b. Kano Quadrant Mapping

The Kano quadrant mapping was conducted in two stages: calculating the better coefficient and worse coefficient, followed by plotting the Kano quadrants. The coefficients were calculated based on the Kano questionnaire results shown in Table 5. The complete results of the better and worse coefficient calculations are presented in Table 6.

Table 6. Better and Worse Coefficient Calculation

Code	Attribute	Better Coefficient	Worse Coefficient
F1	Order Description	0,478	0,521
F2	Order History	0,543	0,565
F3	Freelance Service Recommendation	0,521	0,369
F4	User Help	0,34	0,617
F5	User Guidance	0,347	0,543
F6	Notification / Warning Message	0,363	0,409
F7	Rating / Review	0,357	0,571
F8	Report	0,272	0,63

Based on the calculation results of the better and worse coefficients, each attribute is then mapped into the Kano quadrants to determine the priority categorization. The mapping results of each attribute into the Kano quadrants are presented in Figure 3.

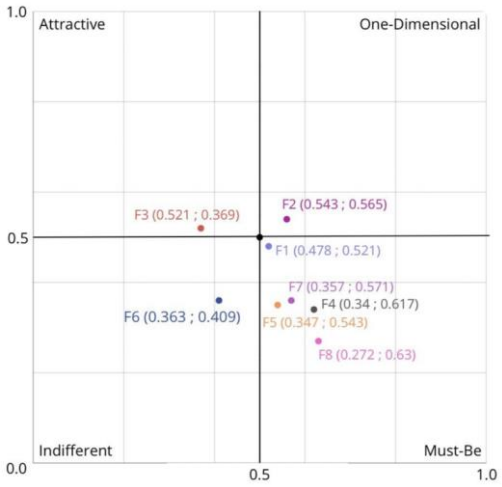

Figure 3. Kano Quadrant Attribute Mapping Results.

c. Kano Classification Evaluation

In the Kano classification evaluation process, product features are categorized into four groups: One-Dimensional, Attractive, Must-Be, and Indifferent. The complete classification results are presented in Table 7.

Table 7. Kano Classification Evaluation

No	Category	Attribute Code
1	Must-Be	F1, F4, F5, F7, dan F8
2	One-Dimensional	F2
3	Attractive	F3
4	Indifferent	F6

1. Must-Be Category

User needs in the Must-Be category represent the fundamental expectations of the system. Therefore, these features must be included without exception. This category is characterized by a low better coefficient and a high worse coefficient. Attributes classified as Must-Be include F1, F4, F5, F7, and F8, indicating that these features are essential components of the application.

2. One-Dimensional Category

User needs in the One-Dimensional category are considered important by users, where higher performance leads to higher satisfaction. This category is characterized by high values for both better and worse coefficients. Attribute F2 falls into this category and is considered an important feature in the application.

3. Attractive Category

The Freelance Service Recommendation feature (F3) was classified as an Attractive attribute because its presence significantly increases user satisfaction, while its absence does not necessarily cause dissatisfaction. This finding indicates that users perceive recommendation functionality as an added value rather than a basic requirement. In freelance platforms, users are often required to spend considerable time searching for suitable services or job opportunities. Therefore, recommendation features can simplify the discovery process, reduce search effort, and improve the overall user experience.

The classification of F3 as an Attractive attribute suggests that users appreciate features that provide additional convenience beyond the core functionality of the platform. While users primarily expect a freelance application to facilitate service transactions and communication between clients and freelancers, recommendation-based support offers extra benefits by helping users identify relevant services more efficiently. This finding indicates that users value personalized assistance in navigating available opportunities, making recommendation features a potential source of competitive advantage for freelance platforms.

4. Indifferent Category

The Notification/Warning Message feature (F6) was categorized as an Indifferent attribute, indicating that its presence or absence has relatively little influence on user satisfaction. Although notification features are commonly implemented in mobile applications, respondents did not perceive them as a critical factor when evaluating a freelance platform. This finding suggests that users place greater emphasis on features that directly support service transactions and job acquisition processes.

The Indifferent classification also indicates that notification functionality may be viewed as a supporting feature rather than a value-adding feature. Unlike recommendation, rating, or reporting features that directly influence user decisions and interactions, notifications primarily serve as an auxiliary mechanism for information delivery. Therefore, while notification features remain useful for system operation, users do not consider them a development priority compared to features that more directly contribute to achieving their objectives within the platform.

These findings are consistent with prior studies on mobile freelance platform development in Indonesia. The Indifferent classification of the Notification feature aligns with Nurdiansyah et al. (2024), who found that Indonesian freelance users prioritize transactional features over auxiliary functionalities. The Attractive classification of the Service Recommendation feature echoes Hanip et al. (2022), who noted that personalized service discovery significantly improves user engagement. Compared to Bachmid et al. (2022), who developed a web-based solution with limited user experience outcomes, the mobile-based approach in this study produced higher user acceptance. Unlike these

prior works, the present study explicitly integrates feature prioritization and iterative development within a unified Kano-RAD framework.

d. Feature Priority Analysis

The feature priority analysis is conducted to determine the final selection of features that align with system objectives and usability, thereby enhancing user experience. Features categorized as Must-Be, One-Dimensional, and Attractive are included in the application development, while features in the Indifferent category, such as Notification/Warning Message (F6), are eliminated due to their low impact on user preferences. The Must-Be features incorporated into the system include Order Description (F1), which provides detailed information about user-created orders; User Help (F4) and User Guidance (F5), which assist users in accessing support and understanding how to use the application; Rating/Review (F7), which enables users to provide feedback in the form of ratings; and Report (F8), which allows users to report problematic workers. In addition, the One-Dimensional feature included in the system is Order History (F2), which helps users track their past orders. Meanwhile, the Attractive feature implemented in the system is Freelance Service Recommendation (F3), which provides personalized service suggestions and recommendations to users. The final selected attributes for further development of the system architecture and application are presented in Table 8.

Table 8. Selected Attributes for System Development.

No.	Category	Feature Type	Code
1.	Information Feature	Order Description	F1
		Order History	F2
2.	Recommendation Feature	Freelance Service Recommendation	F3
3.	Support Feature	User Help	F4
		User Guidance	F5
4.	Evaluation Feature	Rating / Review	F7
5.	Complaint Feature	Report	F8

D. Architecture and Application Development

The feature priority analysis results in selected attributes that serve as the foundation for system development using RAD method. The development process is carried out iteratively through prototyping and Black Box Testing, consisting of five main stages.

a. Business Design

The business design is modeled using a use-case diagram to represent system behavior and interactions with actors. The freelance application system involves two actors: User, who requires services and can only place orders, and Worker/Freelancer, who has registered and can take on jobs by applying to orders created by users. In total, there are 16 use cases interacting with both actors within the system. The identified use cases within the system boundary are il

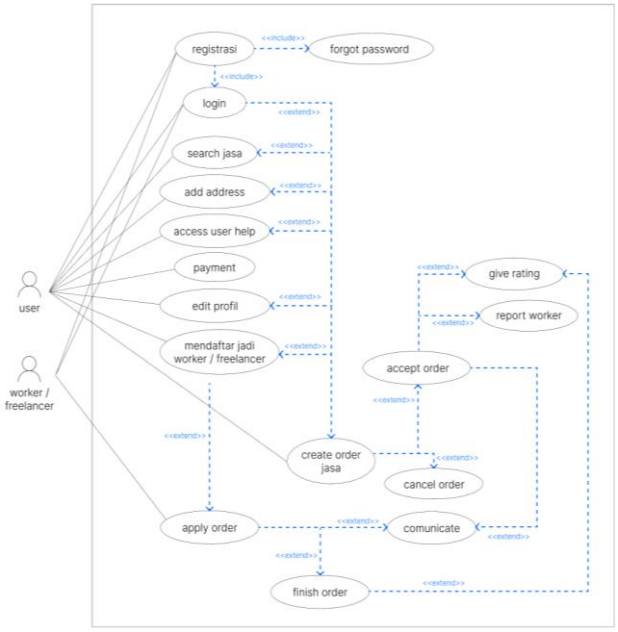

Figure 4. Use Case Diagram of the Freelance Application System

b. Data Design

The data design of the freelance application system is developed using an Entity Relationship Diagram (ERD) to represent entities and their relationships. This ERD serves as the primary guideline for implementing the data model schema in Firebase Firestore. The ERD design is shown in Figure 5.

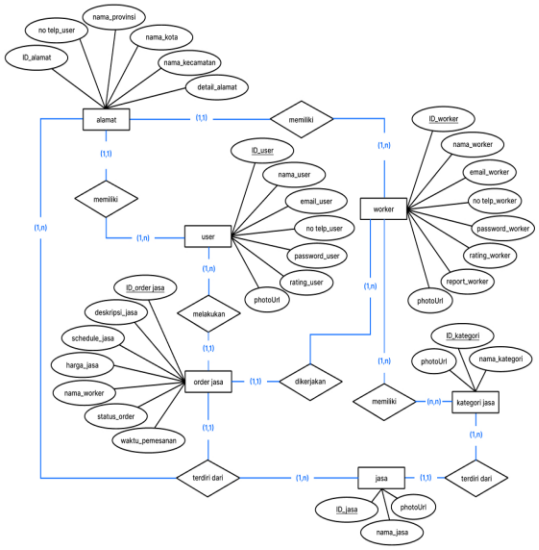

Figure 5. Entity Relationship Diagram of the Freelance Application System

The system consists of five main entities:

1. User
   User entity stores user data such as name, email, phone number, password, rating, and profile photo, and is related to Order Service and Address entities.
2. Worker
   Worker entity stores freelancer data including name, email, phone number, password, rating, and work reports, and is related to Order Service, Address, and Service Category entities.
3. Order Service
   Order Service entity records order details such as service name, description, schedule, price, worker name, order status, and order time, and is related to User, Worker, Address, and Service entities.
4. Service
   Service entity stores available service data, including service name and image, and is related to Service Category and Order Service entities.
5. Service Category
   The Service Category entity groups services based on categories and is related to Worker and Service entities.

c. Prototyping

At the prototyping stage, Black Box Testing is conducted involving four users, covering both system functionality and design aspects, with each undergoing two iterations based on user feedback. Functional testing includes 40 actions, where in the first iteration 19 actions are valid and 21 are invalid; after improvements, all actions become valid in the second iteration. Meanwhile, design testing involves 21 interface pages, with 10 pages validated in the first iteration and 11 not yet valid, until all pages reach a 100% validity score in the second iteration.

d. Construction

In the construction phase, the application system is developed based on the selected features derived from the Kano Model analysis, iterative feedback, and the previously designed system architecture. The following are several implementations of the developed system design.

1. Account Registration and Login
   a. Login
   Login feature provides a form that must be completed by users before accessing the application dashboard. The Login page also includes a registration feature for new users and a “Forgot Password” feature for users who experience difficulties logging in.
   b. Register
   The Register feature provides a form that must be completed by users to create a new account. The registration process includes fields such as name, email address, phone number, and password.

Table 9. Iteration progress results of system functional testing.

No.	Iteration	Valid Results	Not Valid Results
1.	Iteration 1	Access Login, Fill Login Form, Access Register, Fill Register Form, Access Forgot Password, Fill Forgot Password Form, Access Service Categories, Select Service Category, Fill Order Form, Access On-Going Order, Access Rating, Access Order History, Select Order (in Order History), Access Profile, Access Address, Access “Add New Address”, Fill New Address Form, Access Freelancer Registration, Access Job Order.	Submit Login, Submit Register, Submit Forgot Password, Access Search, Fill Search Form, Create Order, Select Order (in On-Going Order), Submit Cancel Order, Submit Accept Order, Access Chat, Submit Rating, Access Report, Submit Report, Submit New Address, Access User Help, Submit Freelancer Registration, Submit Apply Order, Submit Finish Order, Access Rating, Submit Rating.
2.	Iteration 2	Access Login, Fill Login Form, Access Register, Fill Register Form, Access Forgot Password, Fill Forgot Password Form, Access Service Categories, Select Service Category, Fill Order Form, Access On-Going Order, Access Rating, Access Order History, Select Order (in Order History), Access Profile, Access Address, Access “Add New Address”, Fill New Address Form, Access Freelancer Registration, Access Job Order, Submit Login, Submit Register, Submit Forgot Password, Access Search, Fill Search Form, Create Order, Select Order (in On-Going Order), Submit Cancel Order, Submit Accept Order, Access Chat, Submit Rating, Access Report, Submit Report, Submit New Address, Access User Help, Submit Freelancer Registration, Submit Apply Order, Submit Finish Order, Access Rating, Submit Rating.	

Table 10. Iteration Progress Results of System Design Testing.

No.	Iterasi	Valid Results	Not Valid Results
1.	Iteration 1	Account Registration & Login, Order (On-Going Order), Rating, Report, Profile, My Address & Add Address, Apply Order (Worker), Finish Order (Worker), Give Rating (Worker), Job Tracker Worker (History Order)	Category & Service Mapping, Order Form Input, Cancel Order, Accept Order, Communication, My Order, Order History, User Help, Freelancer Registration, Job Order (Worker), Job Tracker Worker (Report).
2.	Iteration 2	Account Registration & Login, Order (On-Going Order), Rating, Report, Profile, My Address & Add Address, Apply Order (Worker), Finish Order (Worker), Give Rating (Worker), Job Tracker Worker (History Order), Category & Service Mapping, Order Form Input, Cancel Order, Accept Order, Communication, My Order, Order History, User Help, Freelancer Registration, Job Order (Worker), Job Tracker Worker (Report).	

2. Category and Service Mapping
The application includes 13 freelance service categories, namely Daily Cleaning, Service, Education, Handyman, Automotive, Electrical, Photography & Videography, IT/Programming, Design, Home Repair, Plumbing/Water Services, Delivery, and Language Translation services. The selection of these categories is based on freelance job classifications from Freelancer.com, particularly those involving services that require direct interaction. The design of the category and service mapping is shown in Figure 6.

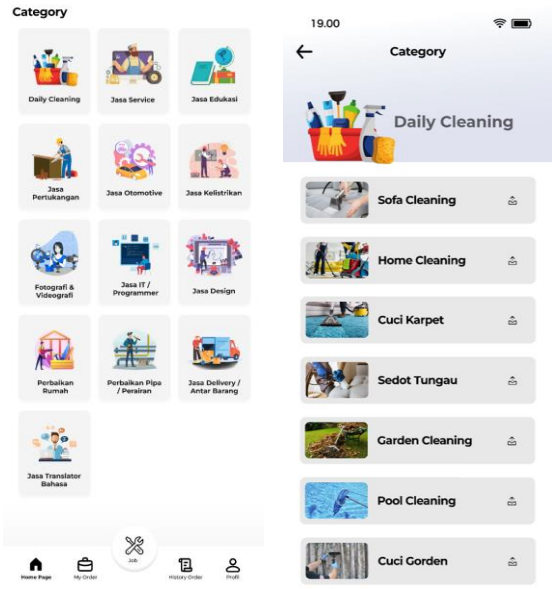

Figure 6. Category and Service Mapping

3. Order Form
The order form is used by users to provide detailed information before placing an order. It consists of two main input fields: a detailed order description and an estimated service completion time. The estimated time field helps workers determine the duration required to complete the task. Users can also specify the service location and set or negotiate the service price according to their preferences. The design of the order form is shown in Figure 7.

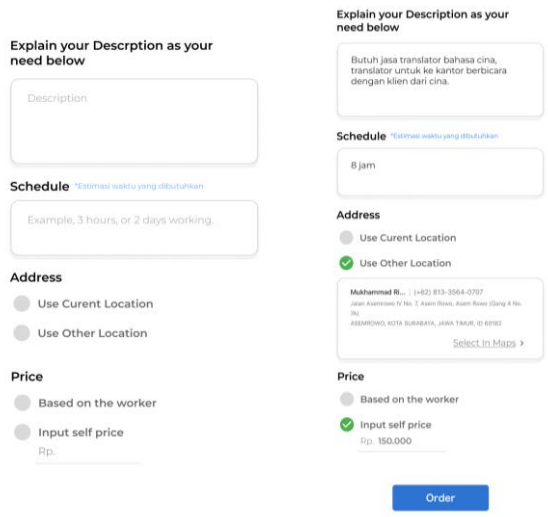

Figure 7. Form Order.

4. On-Going Order
The On-Going Order feature displays detailed information about active orders. Users can cancel an order if needed and select from multiple workers who have applied to the order. Users are able to review and choose workers based on their preferences. The design of the On-Going Order system is shown in Figure 9.
5. Communicate
The Communicate feature enables users and workers to interact with each other. In this application, communication is facilitated by integrating the system with the WhatsApp

platform. The design of the Communicate feature is shown in Figure 10.

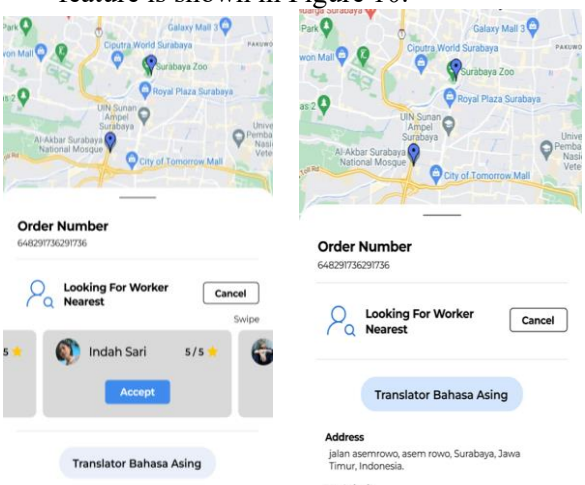

Figure 9. Sistem On-Going Order.

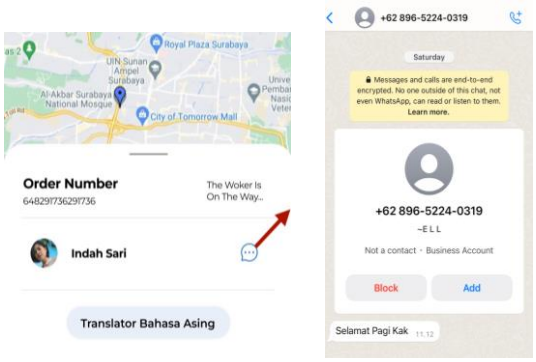

Figure 10. Communicate.

6. Rating
   The Rating feature allows users to provide feedback on the worker's performance by assigning a rating ranging from 1 to 5 stars.
7. Report
   The Report feature allows users to submit complaints regarding workers. This feature enables users to report issues or violations, thereby maintaining system security, quality, and integrity by providing control over problematic situations.
8. Order History
   The Order History feature enables users to view records of their previously created orders.
9. Freelancer Registration
   The Freelancer Registration feature allows users to register as workers or freelancers. The design of this feature is shown in Figure 11.
10. Job Order (Worker)
    Workers can select and apply for available orders by clicking the "apply" button. They can also filter available jobs using the filter menu to find suitable orders. The design of the Job Order feature is shown in Figure 12.

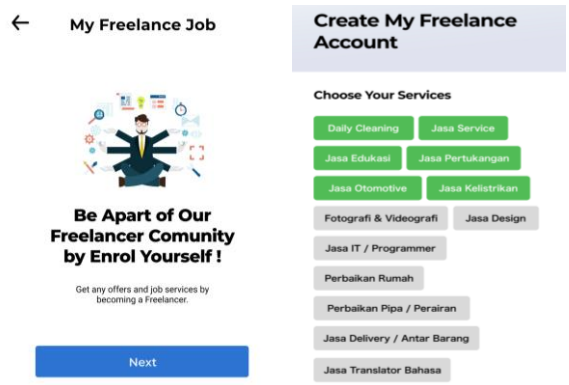

Figure 11. Freelancer Registration.

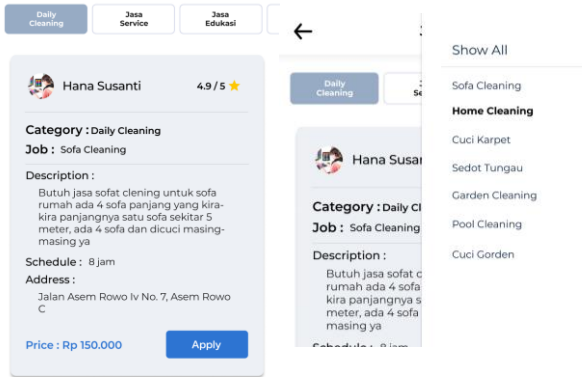

Figure 12. Job Order.

E. Cut-over

The study involved 52 respondents consisting of general users and freelance workers, with the majority (82.7%) aged 21 to 24 years, reflecting the demographic most active in Indonesia's digital freelance ecosystem. General users tended to prioritize features supporting order management and worker selection, while freelance workers emphasized job visibility and transaction transparency. The evaluation was conducted through several stages, including requirement analysis, feature prioritization using the Kano Model, system development using the RAD method, functional testing through Black Box Testing, and UAT. During the UAT process, respondents interacted with the developed application and evaluated its functionality and usability through a structured questionnaire. The assessment focused on several aspects including functional requirement fulfillment, feature implementation, system performance, interface attractiveness, ease of use, application structure, and information presentation.

The UAT questionnaire is designed based on the criteria in Table 2, with score interpretation referring to Table 1. The UAT results show a percentage range between 95% and 96.8%, which falls into the "Very Good" category overall. The average UAT score is 95.9%, indicating that the system has been implemented

with a “Very Good” level of quality. The complete UAT results are presented in Table 11.

Table 11. UAT Testing Results.

Code	Question	Percentage Result
Q1	Does the application meet functional requirements according to your expectations?	96,4%
Q2	Have all system features and specifications been implemented correctly?	96,8%
Q3	Does the application workflow run properly and meet your expectations?	95,6%
Q4	Is the user interface design attractive?	96,2%
Q5	Are the interface and menus easy to understand?	96,2%
Q6	Does the application structure make it easy to use?	95,6%
Q7	Does the information presentation help you find what you need easily?	95%
	Average=	95,9%

The consistently high scores across all evaluation aspects indicate that users positively perceived both the functional and usability aspects of the application. The highest score was obtained for feature implementation accuracy (96.8%), while the lowest score was recorded for information presentation (95.0%). Nevertheless, all evaluation aspects achieved scores above 95%, suggesting a high level of user acceptance. These findings support the overall UAT score of 95.9% and demonstrate that the developed system was well accepted by users and successfully met user expectations.

From a usability standpoint, the highest score for feature implementation accuracy (96.8%) reflects the effectiveness of the two-cycle Black Box Testing process in translating user requirements into functional components. The lowest score for information presentation (95.0%), while still categorized as Very Good, indicates a potential area for improvement in content organization and information hierarchy in future iterations. A similar finding was reported by Zaman et al. (2024), who showed that applying the User-Centered Design (UCD) method effectively improved interface clarity and information presentation through iterative redesign based on user feedback. Overall, scores exceeding 95% across all seven dimensions confirm that the Kano-based feature prioritization combined with RAD iterative development produced a system that effectively meets user usability expectations.

IV. CONCLUSION

The implementation of the Kano Model in feature priority analysis was carried out through four stages to effectively process user requirements and determine the necessary features for the application system. The results of the analysis identified seven priority features, namely Order Description (F1), Order History (F2), Freelance Service Recommendation (F3), User Help (F4), User Guidance (F5), Rating/Review (F7), and Report (F8). The application system was developed using the RAD method based on the prioritized features derived from the Kano Model. During the development process, iterative improvements and user feedback were utilized to refine the system, ensuring that user needs were effectively fulfilled. The UAT results show an average percentage score of 95.9%, indicating that the system has been implemented with a "Very Good" level of quality.

This study contributes to the body of knowledge by demonstrating an integrative Kano-RAD framework that combines feature prioritization with iterative system development, an approach that has been applied separately in prior studies but rarely combined within a single freelance platform development context. Furthermore, the integration of the Kano Model and RAD method supports a development process that is both user-centered and efficient by combining feature prioritization with rapid iterative development. The findings demonstrate that the Kano Model provides practical benefits in prioritizing application features based on their impact on user satisfaction, enabling developers to focus resources on functionalities that deliver the greatest value to users. This approach can serve as a useful reference for future freelance marketplace platforms, particularly in developing applications that are more responsive to user needs, adaptable to changing requirements, and capable of improving overall user satisfaction.

This study is subject to several limitations. The sample of 52 respondents was obtained through purposive sampling within a limited social network, which may affect the generalizability of the feature prioritization results. The communication feature relies on third-party WhatsApp integration, and the backend infrastructure based on Firebase Firestore may face scalability constraints as the user base grows. Additionally, the RAD development process was limited to two iteration cycles, which may not fully capture all edge cases in complex user interaction scenarios.

For future research, it is recommended to enhance the user interface design by benchmarking similar applications in the market and to explore alternative methods for analyzing feature

requirements, as well as to develop more advanced features such as AI-based recommendation systems and trust-enhancing mechanisms to improve personalization, reliability, and user engagement within the platform..

REFERENCES

[1] Abraham, J., Ismail, I. E., Kom, S., & Kom, M. (2021). Unit Testing dan User Acceptance Testing pada Sistem Informasi Pelayan Kategorial Pelayanan Anak. Repository PNJ.

[2] Al Anshari, A. A., & Yuamita, F. (2024). Analisis Tingkat Kepuasan Konsumen Terhadap Produk Kandang Kelinci UMKM Dandellion Menggunakan Metode Kano. Jupiter: Publikasi Ilmu Keteknikan Industri, Teknik Elektro dan Informatika, 2(2), 206–215.

[3] Aristi, N. M., & Pratama, A. R. (2021). Peran Freelance Marketplace dan Media Sosial dalam Online Gig Economy Jasa Profesional. Techno.Com, 20(1), 122–133.

[4] Bachmid, M. A., Ananta, M. T., & Brata, K. C. (2022). Pengembangan Aplikasi Pencarian dan Penawaran Kerja Paruh Waktu untuk Usaha Mikro, Kecil, dan Menengah (UMKM) berbasis Progressive Web App (Studi Kasus Kota Malang). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 6(12).

[5] Hanip, M. I., & Hamid, N. A. (2022). MyFREELANCER App. Applied Information Technology And Computer Science, 3(2), 1130–1143.

[6] Hutabri, E. (2022). Validitas Media Pembelajaran Multimedia Pada Mata Pelajaran Simulasi Dan Komunikasi Digital. Prosiding Seminar Nasional Ilmu Sosial Dan Teknologi (SNISTEK), 296–300.

[7] Irwinansyah, M. I., Tolle, H., & Brata, K. C. (2020). Perancangan Pengalaman Pengguna Aplikasi Pencari Partner Lomba Bagi Mahasiswa Berbasis Mobile Menggunakan Metode Design Thinking. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 4(9), 2843–2850.

[8] Munawir, A., & Nugroho, N. (2023). Penerapan Metode Rapid Application Development Pada Sistem Informasi Monitoring Pelanggaran Siswa. Jurnal Ilmiah Sistem Informasi Akuntansi, 3(1), 69–78.

[9] Natalia, C., & Putranto, G. F. (2023). Kerentanan Kesejahteraan Gig Worker di Indonesia Pascapandemi. Jurnal Ekonomi Indonesia, 12(2), 173–186.

[10] Nugroho, J., Hardian, V., Ismail, D. H., & Dwi Raharjo, J. S. (2023). Economic GIG In Indonesia: Challenges And Opportunities For Gen Z And Milenials. Best Journal of Administration and Management, 2(1), 27–34.

[11] Nurdiansyah, M. I., et al. (2024). Development of Mobile-Based Freelance Services Marketplace Application System. Jurnal Teknik Informatika (JUTIF), 5(5), 1365–1374.

[12] Profita, A., Ifan, A. N., & Burhandenny, A. E. (2022). Penerapan Metode Rapid Application Development (RAD) Untuk Digitalisasi UKM Industri Busana Muslim. Jurnal Rekayasa Teknologi Informasi (JURTI), 6(2), 171–179.

[13] Purwanti, E., & Zaman, B. (2016). Identifikasi Kebutuhan Operasional CRM untuk Monitoring Tugas Akhir. Jurnal Multinetics, 2(2), 70–74.

[14] Reuschke, D., & Wilkins, A. H. (2017). Freelance Working: Mobile Technology And Social Media.

[15] Rif'ah, A., Sitania, F. D., & Gunawan, S. (2020). Designing Lai Pie Product By Using Kano Model And QFD Method. Jurnal Agritechno, 112–119.

[16] Riri, et al. (2022). Pengukuran Kualitas Layanan pada Aplikasi Mobile JKN Menggunakan Metode E-Service Quality dan Model Kano. Coding: Jurnal Komputer dan Aplikasi.

[17] Suwandi, H., Harlinda, H., & Mansyur, S. H. (2022). Implementation of a School Information System Using Rapid Application Development Method. Jurnal Teknik Informatika (Jutif), 3(6), 1501–1512.

[18] Victoria, O., Kaunang, F. J., & Wagiu, E. B. (2022). Component Analysis of Layout Design, Color, and User Interface Control on User Experience in PeduliLindungi Applications. TeIKa, 12(02), 121–133.

[19] Wijaya, J., Saputra, C., Imletta, S. F., Hakim, M. A. A., Pilipus, G. C., & Pribadi, M. R. (2022). Perancangan Aplikasi FindJob Untuk Freelancer Dalam Mencari Pekerjaan Menggunakan Metode Design Thinking. MDP Student Conference, 1(1), 430–435.

[20] Zainah, S., Hamzah, M. L., Rozanda, N. E., & Salisah, F. N. (2023). Analisis Kualitas Layanan E-Commerce Shopee Menggunakan Metode E-Servqual Dan Kano. JATISI (Jurnal Teknik Informatika dan Sistem Informasi), 10(2), 316–328.

[21] Zaman, B., Nayottami, I. B., & Hariyanti, E. (2024). Evaluasi dan Desain Ulang Antarmuka Menggunakan Metode User Centered Design (UCD) pada Aplikasi Kampus Kita Tendik. J-Icon: Jurnal Komputer dan Informatika, 12(1), 91-101.