JURNAL ILMIAH FIFO
Vol. 17 No. 1 (2025) 7-16
DOI: 10.22441/fifo.2025.v17i1.002

P-ISSN 2085-4315
E-ISSN 2502-8332

User Requirements Analysis for Government's Budget Information System Using Kano's Model

Isnen Hadi Al Ghozali[1]*, M. Askar Fathin[2], Andy Rio Handoko[3]

[1,3]Fakultas Teknologi Informasi, Universitas Budi Luhur
[2]Sekolah Vokasi, Universitas Gadjah Mada
Email: [1]2111601163@student.budiluhur.ac.id, [2]muhammadaskar@mail.ugm.ac.id,
[3]andy.handoko@budiluhur.ac.id
Corresponding Author*

(***received*****:** 20-06-24, ***revised*****:** 03-07-24, ***accepted*****:** 19-03-25)

Abstrak

Penelitian ini bertujuan menemukan atribut yang masih memerlukan perhatian berdasarkan analisis model Kano, prioritas kebutuhan perangkat lunak dalam perangkat lunak penganggaran pada Kementerian/ Lembaga. Penelitian ini memprioritaskan atribut kebutuhan perangkat lunak dengan menggunakan Model Kano. Penelitian ini menggunakan aplikasi Kementerian Keuangan sebagai dasar penyusunan kuesioner dengan responden 75 orang PNS. Penelitian ini menyimpulkan bahwa atribut pengembangan perangkat lunak terdiri dari: 15 sub-elemen (termasuk 25 fitur) yang tergolong *one-dimensional* dan empat sub-elemen (termasuk delapan fitur) yang tergolong *Indifferent*. Berdasarkan formula Blauth dan analisis data kontinyu, hasilnya menunjukkan pola *one-dimensional*. Berdasarkan *CS Coefficient*, terdapat sembilan fitur yang diprioritaskan untuk dikembangkan (khususnya proses revisi anggaran). Pengembangan pada dimensi kebutuhan antarmuka dan kebutuhan fungsional berpotensi meningkatkan kepuasan *end user*.

Kata Kunci: *user requirement, software requirements,* pemerintahan, informasi anggaran*,* Model Kano

Abstract

This study finds out what attributes still require attention based on Kano's Model analysis, which has to be used to prioritize software requirements in the government's or agency's budget software. This research prioritizes software requirement attributes using Kano's Model. This research used an application by Ministry of Finance of Indonesia as the basis for preparing a questionnaire distributed to 75 civil servants. This study concludes that the attributes of software development are composed of: 15 sub-elements (including 25 features) classified as one-dimensional and four sub-elements (including eight features) classified as Indifferent. According to the Blauth Formula and continuous data analysis, the result shows a one-dimensional pattern. Based on the CS Coefficient, nine features are prioritized for development (especially the process budget revision). Development on the dimensions of interface requirements and functional requirements has the potential to increase the end user's satisfaction.

Keywords*: user requirement, software requirements, government, budget information, Kano's Model*

1. INTRODUCTION

The digitalization of technology, particularly information technology, in supporting the efficiency of business processes is critical to ushering in the "Society 5.0" era. It triggers automated applications in various sectors, including the government sector. To get a practical application, software requirements are a critical phase of an application's entire development life cycle [1]. Sometimes, developers get frustrated because they need help understanding what users want. It makes it more likely that the project they are working on will fail [2].

It aligns with the results of the 2014 Standish Report study, which states that the biggest challenge to software requirements is Incomplete Requirements (13%) [1]. Based on these facts, developers develop alternative methods for defining user requirements when developing applications. Several methods are often used, such as user stories [2], Scoring Rubrics-Assisted Reading [3], Lean Six Sigma [1], and the Kano Model [4]. Although the user requirement phase does not yet refer to an application development method, there is a user requirement method that refers to an application development method. For example, agile software development generally uses the user story method. The user stories method has the advantage of a simple and easy-to-understand format. However, the weaknesses of this method are that it lacks detail, often triggers

P-ISSN 2085-4315
E-ISSN 2502-8332

ambiguity, and is not suitable for large projects. The Scoring Rubrics-Assisted Reading technique serves as an alternative to address the shortcomings of the user stories method, particularly in identifying weaknesses. Nonetheless, this technique necessitates significant experience and considerable time to complete. The Lean Six Sigma technique offers another, more sophisticated alternative, being data-driven and focused on efficiency. Unfortunately, this technique is highly complex and also requires substantial time for completion. A more moderate alternative method is the Kano Model. The Kano Model identifies qualitative attributes that emphasize user satisfaction and latent needs. The advantage of this method, compared to others, is its ability to prioritize software development features. This capability is especially beneficial for completing large projects within limited time frames. Consequently, this research focuses on the Kano Model to identify prioritized attributes in the development of budgeting software for ministries and institutions.

The Electronic-Based Government System (EBGS) policy, as outlined in Presidential Regulation No. 95 of 2018, realizes the use of automated applications in the field of government. EBGS implementation is an innovation in the bureaucratic process that takes advantage of information and communication technology revolutions. One of the areas that EBGS focuses on is the Performance and Budget Information System. Budgeting, business processes, and performance achievement monitoring are two critical components of government accountability. In general, the Ministry of Finance has made the SAKTI application for managing budgets and the SMART application for keeping track of how sound things are going [5]. However, these two applications are centralized, so application development cannot be carried out by other ministries or institutions. It is a challenge because the definition of application user requirements may need to cover the needs of various end users fully. Even though user requirements are used to form user habits for using digital products [6]], this requirement needs to be reviewed because business processes in the Work Unit (the minor organizational level) still need to be accommodated in the application. For example, the SAKTI application needs to be equipped with features for budget revisions, which are the authority of the leaders of ministries and agencies [7]. Also, the SMART application doesn't have a way to show performance accountability reports (LAKIP) just yet. Also, the SMART application has yet to show performance accountability reports (LAKIP).

User requirements are based on needs and expectations, which are not immutable. It is, therefore, natural that the requirements of students will change over time, reflecting their internal expectations [8]. Software requirements describe the functions to be designed and developed in a given software system [9]. A method of defining user requirements that provides convenience and objectivity is the Kano Model. The Kano model is an instrument developed by Professor Noriaki Kano to identify and classify quality attributes based on their effect on customer satisfaction into must-be, attractive, one-dimensional, inverse, and indifference attribute categories [10]. The Kano Model is indeed often associated with assessing service quality [11]; in the study [10], the Kano Model is one of the methodologies that will be widely discussed in future research. The Kano Model classifies attributes of software requirements that can meet the needs of end users. The Kano Model classifies service attributes into five categories. In the must-be category or basic needs, the user becomes dissatisfied if the attribute is low. Things to note: Customer satisfaction remains the same despite high attribute performance. Different things are found in the one-dimensional or performance needs category; customer satisfaction is positively correlated with attribute performance, so high attribute performance will lead to high customer satisfaction. For the attractiveness or excitement needs category, an increase in attribute performance triggers a high increase in customer satisfaction.

The Kano Model can help developers identify more precise user requirements, reducing human errors. Developers need methods to prevent and detect project faults to improve software quality [11]. The Kano Model is a method that assists developers in preventing errors by identifying software requirement specifications based on user preferences, reducing the risk of misunderstandings regarding environment, features, or non-functional requirements. In previous studies, the Kano Model was used more to measure service quality [12] and mobile application user requirements in the private sector [13] [14] [15] [16]. Our study will analyze the user requirements of the government's performance and budget information system using Kano's Model.

2. METHODS

This study uses a case study method by examining the features of the Performance and Budget Information System application used by the Indonesian Financial Transaction Reports and Analysis Center (INTRAC) as a reference for further researching the software requirements specification (SRS) needed for application development. We use this application to identify and prioritize software requirements that can be used for application development that can connect the SAKTI Application and the SMART Application. This research design consists of five stages, as follows: a) Identify the research question; b) Collect data; c) Measure variables; d) Classification and Preliminary Prioritization by Kano's Model; and e) Prioritization by the CS Coefficient Matrix [17]. Figure 1 shows the research framework used in the research.

JURNAL ILMIAH FIFO
Vol. 17 No. 1 (2025) 7-16
DOI: 10.22441/fifo.2025.v17i1.002
P-ISSN 2085-4315
E-ISSN 2502-8332

The population in this study were employees who worked in the secretarial field at the 82 ministries and agencies. The sample selection was determined using purposive sampling. Purposive sampling is a sampling technique with specific considerations. The balance of the sample in this research consists of respondents who meet the following criteria: a) Employees who work in ministries or agencies within the scope of the Directorate of Budget for Political, Legal, Defense and Security Affairs and the Budget Section of the State General Treasurer b) Employees who work in planning and budgeting or who need performance and budget information. The selection of samples using these criteria is based on the premise that employees whose areas of work are related to planning, budgeting, performance, and budget information are more objective in determining the attributes that need to be prioritized in developing budgeting software. This is because such samples are better equipped to identify the necessary attributes and areas for development compared to existing software. We targeted 75 respondents in this study.

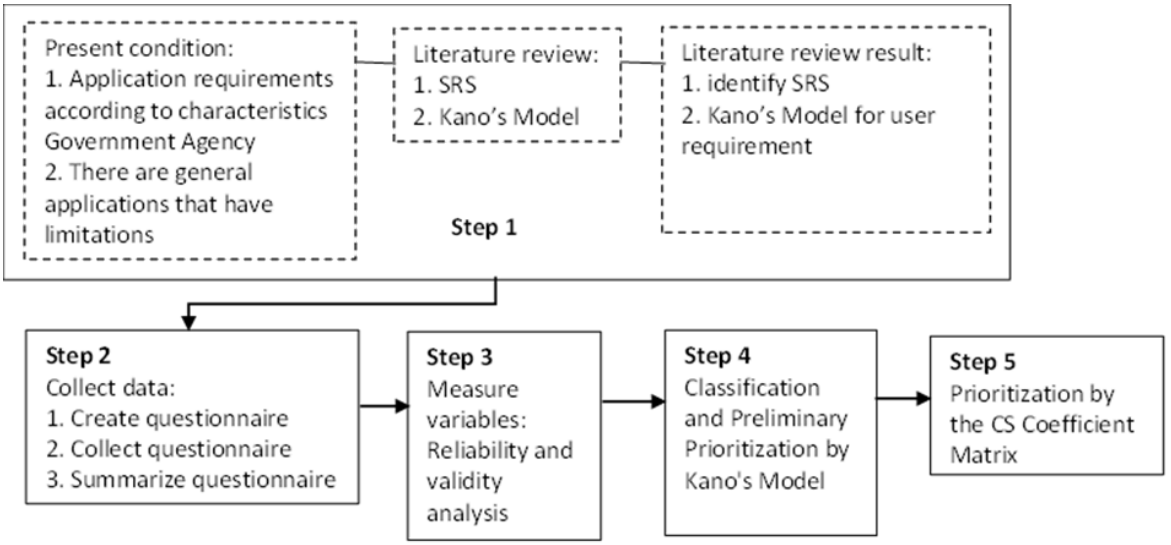


Figure 1. Research Framework

Variables used in this research as measurement instruments are based on IEEE Std. 830-1993. Enhancement of standard software requirement specifications by [18], [19], and [16] Descriptive statistics will be applied to all variables in this research in order to get a general description and a detailed explanation of the phenomena to be analyzed.

The Kano model is a method for measuring user satisfaction using a two-dimensional curve [15]. The two-dimensional curve developed in the Kano Model is intended to overcome the limitations of the quality of the linear model [20]. The two-dimensional curve developed by Kano consists of linear and non-linear curves. According to Kano's perspective, the curve is described in this form, which states that consumers perceive product or service attributes differently and tend to differentiate product functions [17]. Kano's Model Analysis is a method that may help designers solve potential trade-offs by showing which features maximize user satisfaction. The Kano Model questionnaire quantitatively includes questions about functional requirements (FRs). Each FR will consist of a pair of questions: functional and dysfunctional. The answer choices for this question are: 1) like it that way; 2) It must be that way; 3) neutral; 4) can live with it that way; and 5) dislike it that way. The results of the respondents' answers will be codified using Table 1.

After codification, user satisfaction (CS) and user dissatisfaction (DS) are calculated using the formula:

$$CS_i = \frac{f_A + f_O}{f_A + f_O + f_M + f_i} \quad (1)$$

$$DS_i = \frac{f_O + f_M}{f_A + f_O + f_M + f_i} \quad (2)$$

with:
f_A : Functional Attractive)
f_O : Functional One-Dimensional
f_i : Functional Must-Be
f_M : Functional Indifferent

P-ISSN 2085-4315
E-ISSN 2502-8332

Table 1. Evaluation matrix for Kano's Model

Functional	Dysfunctional				
	1. Like	2. Must-be	3. Neutral	4. Live with	5. Dislike
1. Like	Q	A	A	A	O
2. Must-be	R	I	I	I	M
3. Neutral	R	I	I	I	M
4. Live with	R	I	I	I	M
5. Dislike	R	R	R	R	Q

A, attractive; O, one dimensional; R, reverse; M, must be; I, indifferent; Q, questionable

3. RESULTS AND DISCUSSION

3.1. Step 1: Identify the research question

Based on observations, the software for budgeting needs and performance information is currently specialized at the national level. It still needs to cover needs at the Government Agency level. Application requirements according to the characteristics of the Government agency. In addition, there are general applications that have limitations. Existing applications are built with separate databases, so data synchronization is a problem that often arises. We utilize the applications used by Ministry of Finance of Indonesia from 2019 to 2021 as research objects. The research questions are as follows: What software attributes need to be prioritized for development based on the Kano Model? Additionally, which dimensions should be developed to increase user satisfaction?

3.2. Step 2: Collect data

We have sent questionnaires to 75 civil servants. The backgrounds of the respondents were collected and analyzed with regard to gender, age, educational background, job functions, and work experience. Figure 2 presents the backgrounds of the respondents.

Figure 2 (a) shows that the majority of respondents were aged 35–44 (42.7%) and held a bachelor's degree (48%). This indicates that most of the respondents are in the productive phase of their careers. Regarding job experience, the distribution of respondents is almost even, with the highest frequency at 11–15 years (28%). Respondents with this job experience have typically used various performance and budget information application versions. As for the job function, most respondents are in budgeting (44%). The proportion of job functions related to high budgeting will provide a more realistic picture of the application that needs to be developed. It is because the use of the application will be more closely related to the budget. The majority of respondents are employed at the Ministry of Finance (33.3%) and INTRAC (29.3%). This distribution suggests that the respondents have work backgrounds related to planning, budgeting, or performance information.

JURNAL ILMIAH FIFO
Vol. 17 No. 1 (2025) 7-16
DOI: 10.22441/fifo.2025.v17i1.002

P-ISSN 2085-4315
E-ISSN 2502-8332

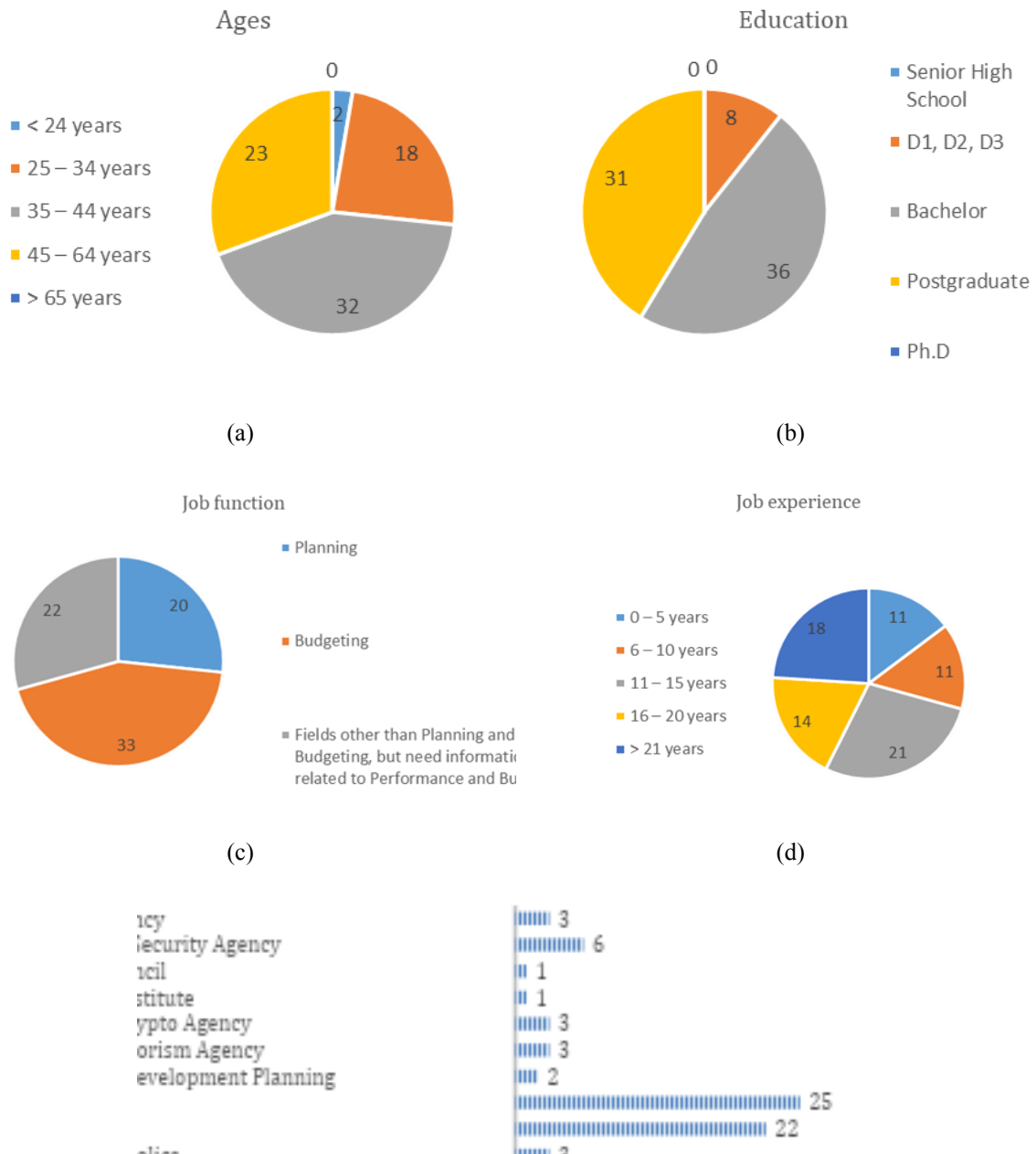


Figure 2. Respondent's background: (a) Ages; (b) Education; (c) Job function; (d) Job Experience; (e) Respondent's ministry or agency.

3.3. Step 3: Measure variables

The questionnaire filled out by the respondent is scored according to the evaluation matrix for Kano's Model. The ordinal scale used is M = 6, O = 5, A = 4, I = 3, R = 2, Q = 1. The research variables are measured based on Confirmatory Factor Analysis (CFA), a technique in which a priori, theoretical, and conceptual indicators are known or determined, and variables that include them are entered into the indicator. A validity test was carried out to measure the validity of the data obtained with the questionnaire instrument. Data is declared valid if the

P-ISSN 2085-4315
E-ISSN 2502-8332

outer loading is greater than 0.7 for each indicator [17]. If there are indicators with an outer loading below 0.7, these indicators will be removed from the model. The test is continued with a reliability test to determine to what extent the measurement results using the same object will produce the same data. Data is declared reliable if 1) Cronbach's Alpha value is greater than 0.6, 2) all outer loading is greater than 0.7, and 3) the AVE value is greater than 0.5.

Based on the initial test using smartPLS 4, indicators show an outer loading of less than 0.7. The indicators are B.1.2, B.1.4, B.1.8, B.2.6, B.2.7, B.2.10, B.2.13, B.3.3, B.3.4, B.3.11, B.3.13, B.3.14, B.4.1, B.4.3, B.4.4, B.5.1, B.5.2, and B.5.6. After the indicator is dropped, the results of the validity and reliability tests are shown as follows:

Table 2. Confirmatory Factor Analysis Result

Element	Indicators	Outer Loading	Cronbach's alpha	Average variance extracted (AVE)
B1. Interface Requirement			0.828	0.592
Hardware interface	B.1.2	0.765		
User Interface	B.1.4	0.738		
Software interface	B.1.6	0.701		
Communication interface	B.1.9	0.817		
	B.1.10	0.82		
B2. Functional Requirement			0.96	0.717
User role & Description process	B.2.2	0.782		
Data reference	B.2.3	0.751		
Process Term of Reference (TOR) and Budget-estimate Plan	B.2.5	0.838		
	B.2.6	0.835		
Process Budget revision	B.2.8	0.739		
	B.2.9	0.884		
	B.2.10	0.881		
Budget revision monitoring	B.2.11	0.863		
	B.2.12	0.928		
	B.2.13	0.945		
	B.2.14	0.843		
B3. Non Functional Requirement			0.864	0.595
Performance	B.3.1	0.714		
	B.3.2	0.868		
Memory Limitation	B.3.3	0.736		
Interface	B.3.13	0.796		
Data limitation	B.3.14	0.736		
	B.3.15	0.77		
B4. Software System Attributes			0.868	0.559
Reliability	B.4.1	0.714		
	B.4.2	0.737		

P-ISSN 2085-4315
E-ISSN 2502-8332

Maintainability	B.4.5	0.753		
	B.4.6	0.704		
Security	B.4.7	0.734		
	B.4.8	0.705		
Portability	B.4.9	0.872		
B5. Design Limitation			0.848	0.766
Platform limitation	B.5.1	0.855		
	B.5.2	0.892		
Architecture limitation	B.5.3	0.879		

3.4 Step 4: Classification and Preliminary Prioritization by Kano's Model

Data considered valid is then classified according to the Kano Model. Based on discrete output calculations, four dimensions are categorized as one-dimensional, and one dimension is categorized as indifferent. If broken down in more detail, 15 sub-elements (including 25 features) are classified as one-dimensional, and four sub-elements (including eight features) as indifferent. The one-dimensional category implies that end-user satisfaction will increase when software features become more abundant. The vital thing to note is how to make user requirements explicit in the SRS, considering that this document is a link between the end user's vague concept and the engineering team's requirements [21]. It makes the software deployment activity complex, requiring the end user's direct participation in software modeling to obtain an integrated solution to the end user's needs [22].

The results of this classification will be the basis for helping developers identify features that need to be prioritized. Table 3 show this result. [23] suggest that developers should develop features that are categorized as must-be (M), one-dimensional (O), and attractive (A). Developing these features will add value and product quality and make products more competitive [24]. Meanwhile, the indifferent category can be ignored. In this study, features that fall into the "indifferent" category are non-functional features and system attributes. Even though these features are not the top priority, they must still be there because they are the minimum standards that software must meet. It indicates that the end user believes these features should be present, so they do not draw attention to themselves.

Table 3. Discrete Output

Element	M	O	A	I	R	Q	Category
B1. Interface Requirement	4%	47%	31%	14%	0%	2%	One dimensional
B2. Functional Requirement	6%	62%	18%	13%	0%	1%	One dimensional
B3. Non Functional Requirement	6%	32%	19%	38%	1%	3%	Indifferent
B4. Software System Attributes	5%	42%	17%	32%	2%	2%	One dimensional
B5. Design Limitation	4%	53%	25%	16%	0%	2%	One dimensional

3.5. Step 5: Prioritization by the CS Coefficient Matrix

According to the classification of this Kano model, we calculated the CS Coefficient Matrix using the "Blauth Formula" and the user satisfaction coefficient. The CS range is 0.46-0.86, with a mean of 0.71. The most considerable CS value is in the Process Budget Revision feature. The smallest CS value in the Data Limitation feature The range of DS values is (-77) to (-32), with a mean of -0.56. The most considerable DS value is in the Interface-Non-Functional Requirement feature. The smallest DS value is in the Process Budget Revision feature. The calculation of the user satisfaction coefficient shows a one-dimensional pattern.

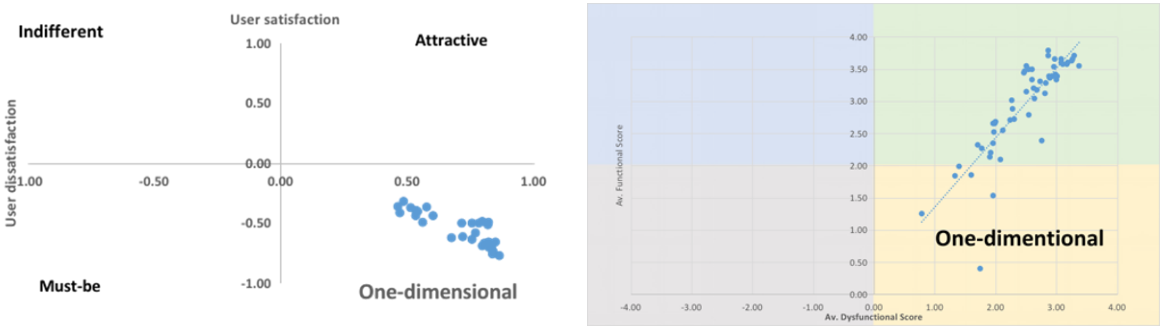
We compare these results with continuous data analysis. First, the data is transformed based on Table 4. The continuous (functional) value range is 2.09–3.71, with a mean of 3.17. The process budget revision feature provides the most functional (continuous) value. The smallest continuous (functional) value in the data

P-ISSN 2085-4315
E-ISSN 2502-8332

limitation feature The continuous (dysfunctional) value range is 1.71 – 3.29, with a mean of 2.62. The most considerable continuous (dysfunctional) value is in the Process Budget Revision feature. The Security feature's smallest continuous (dysfunctional) value Continuous output shows a one-dimensional pattern.

Table 4. Transform matrix from discrete to continuous

	Like it	Expect it	Neutral	Tolerate it	Dislike it
Discrete	4	3	2	1	0
Continuous (Functional)	4	2	0	-1	-2
Continuous (Dysfunctional)	-2	-1	0	2	4



(a) (b)

Figure 3. Graph plot for (a) CS Coefficient and (b) Continuous Output

Evaluation using the user satisfaction coefficient and continuous data analysis shows a one-dimensional pattern. It means that to increase user satisfaction, it is necessary to embed more features. It leads us to ask which features should be prioritized, considering that prioritizing needs takes work, requiring a deep understanding of each requirement's criticality with different factors [25]. In determining priority needs, three main factors are important to note: value, cost, and risk [26]. In determining priority, these three things are the primary considerations: features that provide the most excellent value, the lowest cost, and the lowest risk are, of course, the ideal features to get top priority.

Based on the calculation, nine features meet the criteria for being prioritized. These features are in the interface requirements and functional requirements dimensions. The biggest priority is the Process Budget Revision feature. Respondents pay the most attention to this feature because ongoing business processes are closely related to the need to revise the budget. Budget revisions are needed as a policy response to the dynamic business environment.

4. CONCLUSIONS

Based on the CS Coefficient, nine features are prioritized for development (especially the process budget revision). Development on the dimensions of interface requirements and functional requirements has the potential to increase the end user's satisfaction. Determining priority needs still considers value, cost, and risk [26].

We suggest five prominent features on the functional requirements dimension to be developed in the performance and budget information system software. It is based on the CS Coefficient for this dimension, which has the highest average value compared to other features. For further research, we recommend making a prototype based on the features identified in this study. It is intended to test further whether there has been a change in user preferences since the application was developed.

REFERENCES

[1] N. M. Salleh and P. N. E. Nohuddin, “Optimization of software requirement process: An integrated conceptual model of lean six sigma and requirement planning,” *IRASE*, vol. 10, no. 2, pp. 125–133, Dec. 2019, doi: 10.1556/1848.2019.0015.

[2] D. Hallmann, "'I Don't Understand!': Toward a Model to Evaluate the Role of User Story Quality," in *Agile Processes in Software Engineering and Extreme Programming*, V. Stray, R. Hoda, M. Paasivaara, and P. Kruchten, Eds., Cham: Springer International Publishing, 2020, pp. 103–112. doi: 10.1007/978-3-030-49392-9_7.
[3] E. O. C. Mkpojiogu, N. LailyHashim, and A. Hussain, "Enhancing the Quality of Software Requirements Artifacts with Scoring Rubrics-Assisted Reading Technique," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 8S, pp. 467–474, 2019.
[4] K. Xu, Y. V. Chen, L. Zhang, and L. Rui, "Improving Design Software Based On Fuzzy Kano Model: A Case Study of Virtual Reality Interior Design Software.," *The Design Journal*, vol. 22, no. sup1, pp. 1983–1992, Apr. 2019, doi: 10.1080/14606925.2019.1594923.
[5] Directorate General of Budget, "SISTEM MONITORING DAN EVALUASI KINERJA TERPADU KEMENTERIAN KEUANGAN." Ministry of Finance, 2019. [Online]. Available: http://www.anggaran-old.kemenkeu.go.id/dja/edef-konten-view.asp?id=1449
[6] S. Santini *et al.*, "User Requirements Analysis of an Embodied Conversational Agent for Coaching Older Adults to Choose Active and Healthy Ageing Behaviors during the Transition to Retirement: A Cross-National User Centered Design Study," *IJERPH*, vol. 18, no. 18, p. 9681, Sep. 2021, doi: 10.3390/ijerph18189681.
[7] M. R. Alfarisi, "Overview: Sekilas Tentang Sakti," Jakarta, Aug. 31, 2022. [Online]. Available: https://klc2.kemenkeu.go.id/kms/knowledge/overview-sekilas-tentang-sakti-ae981c88/detail/
[8] P. Madzík, P. Budaj, D. Mikuláš, and D. Zimon, "Application of the Kano Model for a Better Understanding of Customer Requirements in Higher Education—A Pilot Study," *Administrative Sciences*, vol. 9, no. 1, p. 11, Jan. 2019, doi: 10.3390/admsci9010011.
[9] F. Bozyiğit, Ö. Aktaş, and D. Kılınç, "Linking software requirements and conceptual models: A systematic literature review," *Engineering Science and Technology, an International Journal*, vol. 24, no. 1, pp. 71–82, Feb. 2021, doi: 10.1016/j.jestch.2020.11.006.
[10] Q. Meng and J. Dong, "Future Direction and Visual Analysis of Kano Model: A Literature Review," *JSSM*, vol. 11, no. 04, pp. 399–413, 2018, doi: 10.4236/jssm.2018.114028.
[11] V. Anu, W. Hu, J. C. Carver, G. S. Walia, and G. Bradshaw, "Development of a human error taxonomy for software requirements: A systematic literature review," *Information and Software Technology*, vol. 103, pp. 112–124, Nov. 2018, doi: 10.1016/j.infsof.2018.06.011.
[12] F. Rindani and S. Puspitodjati, "Integration of Webqual Method to Importance Performance Analysis and Kano Model to Analyze System Quality of E-Government: Case Study LAPOR!," *J. Sistem Inf. (J. Inf. Sys.)*, vol. 16, no. 2, pp. 1–17, Oct. 2020, doi: 10.21609/jsi.v16i2.937.
[13] T. Choedon and Y.-C. Lee, "Classification and Evaluation of Service Requirements in Mobile Tourism Application Using Kano Model and AHP," *The Journal of Information Systems*, vol. 27, no. 1, pp. 43–65, Mar. 2018, doi: 10.5859/KAIS.2018.27.1.43.
[14] S. Shahidi, "Comparing the effectiveness of conventional and Kano model questionnaire for gathering requirement of online bus reservation system," *ijirss*, vol. 3, no. 1, pp. 27–32, Mar. 2020, doi: 10.53894/ijirss.v3i1.30.
[15] R. A. S. Putra and Priyanto, "Kano Model Analysis of Android Apps Quality from End User's Preferences," *J. Phys.: Conf. Ser.*, vol. 1737, no. 1, p. 012016, Jan. 2021, doi: 10.1088/1742-6596/1737/1/012016.
[16] K. A. Kamaruddin and N. A. Roslan, "Using Kano Model to Prioritize Requirements for UiTM Share Ride Mobile Application Development," *IJEAT*, vol. 9, no. 1, pp. 1752–1757, Oct. 2019, doi: 10.35940/ijeat.A2653.109119.
[17] M.-T. Lu, H.-P. Lu, and C.-S. Chen, "Exploring the Key Priority Development Projects of Smart Transportation for Sustainability: Using Kano Model," *Sustainability*, vol. 14, no. 15, p. 9319, Jul. 2022, doi: 10.3390/su14159319.
[18] C. Rawis, S. D. S. Karouw, and S. R. U. A. Sompie, "Software Requirement Specification Academic Information System of Sam Ratulangi University," *Jurnal Teknik Elektro dan Komputer*, vol. 10, no. 2, pp. 107–118, 2021.
[19] D. Januarita and W. A. Prabowo, "Software Requirement Specification Sistem Informasi Manajemen Rumah Makan Berdasarkan ISO/IEC/IEEE 29148-2018," *SISFOKOM*, vol. 9, no. 2, pp. 215–221, Jul. 2020, doi: 10.32736/sisfokom.v9i2.872.
[20] Y. Qu, M. Xinguo, S. Qiu, Z. Liu, X. Zhang, and Z. Hou, "Integrating fuzzy Kano model and fuzzy analytic hierarchy process to evaluate requirements of smart manufacturing systems," *Concurrent Engineering*, vol. 27, no. 3, pp. 201–212, Sep. 2019, doi: 10.1177/1063293X19845137.

P-ISSN 2085-4315
E-ISSN 2502-8332

[21] U. Ahmed, Amjad Farooq, and Tayyaba Farhat, "ReqSpecOnto: Investigating Explicit Software Requirements Specification," *ICR*, vol. 1, no. 2, pp. 44–70, Dec. 2021, doi: 10.32350/icr.0102.03.
[22] J. C. Pereira and R. de F. S. M. Russo, "Design Thinking Integrated in Agile Software Development: A Systematic Literature Review," *Procedia Computer Science*, vol. 138, pp. 775–782, 2018, doi: 10.1016/j.procs.2018.10.101.
[23] D. Lupita Sari and C. Niswatin, "Functional Requirement on Proofreading System," *JSW*, vol. 14, no. 5, pp. 192–199, May 2019, doi: 10.17706/jsw.14.5.192-199.
[24] A. Hussain, E. O. C. Mkpojiogu, and M. Kutar, "The Impact of Software Features' Perceived Importance on the Perceived Performance of Software Products' Quality Elements," *j comput theor nanosci*, vol. 16, no. 5, pp. 2135–2140, May 2019, doi: 10.1166/jctn.2019.7863.
[25] S. Albuga and Y. Odeh, "Towards Prioritizing Software Business Requirements in Startups," in *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, Amman: IEEE, Jul. 2018, pp. 257–265. doi: 10.1109/CSIT.2018.8486216.
[26] Y. He and J. Zhong, *Improve Requirement Prioritization By End-user Demands : Model Building and Evaluation*. 2021. Accessed: Nov. 19, 2022. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:bth-22254