VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v11i2.6576

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

RE-DESIGNING JAKLINGKO APPS UI/UX USING AGILE REQUIREMENT ENGINEERING APPROACH

Deki Satria[1]*; Qilbaaini Effendi Muftikhali[1]; Dea Wemona Rahma[1]; Dimas Bayu Arkaan[1]; Zain Ammar Falih[1]

Information System[1]
Telkom University, Jakarta, Indonesia[1]
Telkomuniversity.ac.d[1]
dekisatria@telkomuniversity.ac.id*

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract*—Public transportation has become a staple in a lot of countries, including Indonesia. As the largest city in Indonesia, is trying to accommodate the dense traffic in Jakarta by implementing various types of public transportation, one of which is the Bus Rapid Transit (BRT). BRT has its own application called Jaklingko, which the commuter uses to gain information about the BRT. Unfortunately, this application has bad reviews in the app store. This research tried to redesign the UI/UX of this application using prototyping and the System Usability Scale (SUS) as tools for agile requirement engineering tools. In Agile requirements usually conducted the same as traditional which is using interview or observation. But, using this method proved to be time consuming. Therefore this research tried to incorporate prototyping and SUS into the requirements gathering process. After the requirements are collected, the next phase is redesigning the application based on the gathered requirements. From the research conducted, the main pain point of the responses is how much information is given in the apps. This research also found that prototyping and SUS could be used to gather requirements, but they will depend heavily on the test case being used. Therefore, it is not suitable for stand alone gathering tools but good as a confirmation tool.*

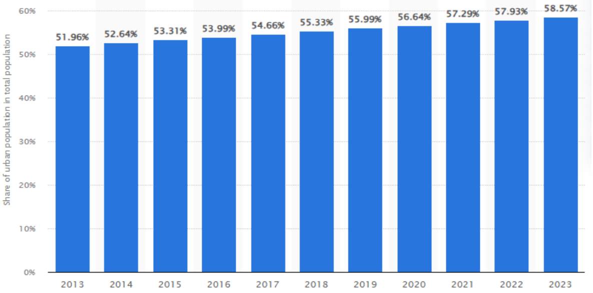
Keywords*: agile requirements engineering, prototyping, software requirements, software usability scale.*

Intisari—*transportasi publik, menjadi sebuah kebutuhan dalam berbagai negara termasuk Indonesia. Jakarta sebagai sebuah kota terbesar di Indonesia berusaha untuk mengakomodir padatnya lalu lintas di Jakarta dengan menerapkan berbagai macam transportasi umum, salah satunya Bus Rapid Transit (BRT). BRT memiliki aplikasi sendiri yaitu aplikasi Jaklingko untuk menyediakan informasi bagi pengguna mengenai BRT tersebut. Sayangnya, aplikasi ini memiliki review yang buruk di appstore. Penelitian ini berusaha untuk merancang ulang aplikasi Jaklingko menggunakan Pendekatan Agile Requirements Engineering dengan memanfaatkan prototyping dan Software Usability Scale (SUS) sebagai peralatan untuk menggali kebutuhan pengguna. Dari penelitian yang dilakukan, ditemukan bahwa kendala utama dari pengguna dalam penggunaan aplikasi ini adalah informasi yang diberikan tidak terlalu informatif. Selain itu dalam penelitian ini juga ditemukan bahwa penggunaan prototype sebagai alat dalam penggalian kebutuhan pengguna, namun lebih sesuai untuk mengkonfirmasi. Hal ini dikarenakan kebutuhan yang didapatkan sangat bergantung pada kualitas test case yang dikembangkan.*

Kata Kunci*: agile requirements engineering, kebutuhan sistem, prototyping, software usability scale.*

INTRODUCTION

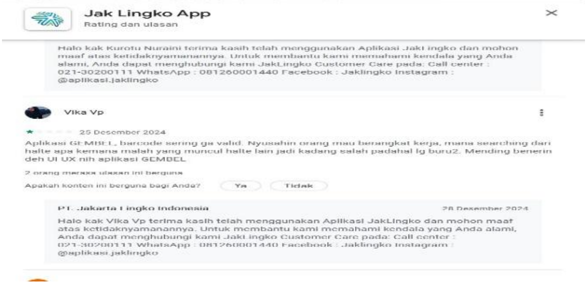
Urbanization has become a problem in a lot of countries, including Indonesia. Statista published that from 2015 to 2023, urbanization kept on growing. Where in 2023, the growth exceeded 58% [1]. Figure 1 shown the urbanization growth in Indonesia. This urbanization became a problem for the city. The problems include traffic congestion, high logistic costs, social injustice, housing, water, waste and other problems [2], [3]. One of the main problems caused by urbanization is traffic congestion. Traffic congestion mainly caused by the private ownership if transportation. To counter this problem, Many Countries depend on Public Transportation. The mass mainly uses public transportation for their public commuting transportation of choices. Jakarta is one of the most crowded cities in Indonesia. Based on data from databoks.com, Jakarta became Indonesia's second most congested city [4]. Therefore, Jakarta has also developed its public transportation systems, which are BRT (Bus Rapid Transit), MRT (Mass Rapid Transit), LRT (Light Rapid Transit), and Commuter Line. Based on data gathered by the Indonesian Central Bureau of Statistics, the number of passengers on public transportation increased yearly. The data shows that MRT and BRT have more passengers in 2024 compared to 2023, where the increase is 3,54% for MRT and 6,83% for BRT [5].



Source: (Aaron O'Neill, 2025)

Figure 1 Indonesia Urbanization Growth Rate

BRT is a public transportation model that combines a bus's flexibility and a train's effectiveness with dedicated roads and high capacity. BRT became one of the best transportation choices in many cities [6].



Halo kak Kurotu Nuraini terima kasih telah menggunakan Aplikasi JakLingko dan mohon maaf atas ketidaknyamanannya. Untuk membantu kami memahami kendala yang Anda alami, Anda dapat menghubungi kami JakLingko Customer Care pada: Call center : 021-30200111 WhatsApp : 081260001440 Facebook : Jaklingko Instagram : @aplikasi.jaklingko

Vika Vp

★ 25 Desember 2024

Aplikasi GEMBEL, barcode sering ga valid. Nyusahin orang mau berangkat kerja, mana searching dari halte apa kemana malah yang muncul halte lain jadi kadang salah padahal lg buru2. Mending benerin deh UI UX nih aplikasi GEMBEL

2 orang merasa ulasan ini berguna

Apakah konten ini berguna bagi Anda? Ya Tidak

PT. Jakarta Lingko Indonesia 28 Desember 2024

Halo kak Vika Vp terima kasih telah menggunakan Aplikasi JakLingko dan mohon maaf atas ketidaknyamanannya. Untuk membantu kami memahami kendala yang Anda alami, Anda dapat menghubungi kami JakLingko Customer Care pada: Call center : 021-30200111 WhatsApp : 081260001440 Facebook : Jaklingko Instagram : @aplikasi.jaklingko

Source: (Appstore Reviews, 2024)

Figure 2 Jaklingko Application Reviews

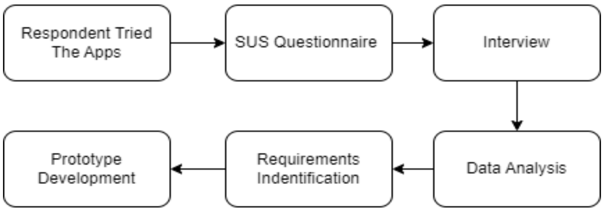
BRT, as one of the public transportation mediums in Jakarta, also provides its passenger with applications that ease their use. This Application is called Jaklingko. This Application gives users features such as a nearby terminal station and payment, which could help a lot of first-time passengers who do not know where to go. Unfortunately, based on the Play Store reviews and ratings, it only gets a 2.4 from 5 stars rating. This low rating shows user dissatisfaction with the apps. Therefore, Jaklingko must change or add its features to increase user satisfaction. The sample of Jaklingko review can be seen in Figure 2. This research tried to find the requirements to re-engineer the apps by finding new designs or other requirements. In this research, we will use an agile requirements engineering approach. Agile requirement engineering is an approach in agile software development dedicated to requirements engineering. This engineering includes the process of understanding, communicating, and managing the requirements of the product [7]. Agile Requirement Engineering has no dedicated approach to conducting the requirement engineering process.

Therefore, many approaches are used in agile requirements engineering, such as MVP, Prototyping, Survey, FGD, test scenarios or stories, qualitative approaches, or customized approaches like DiCOT, ARAM, FlexRQ, or another approach [8], [9]. Using a correct approach in requirements engineering will ensure that the final product meets the user's needs. In contrast, an inadequate approach will result in many requirements not meet [9], [10]. This research also tried to incorporate the System/Software Usability scale as a tool to gather requirements in the prototyping process. The system/Software Usability Scale is a tool used to evaluate a system's usability for a user, which is part of the user acceptance process (UAT) [11]–[13]. In our previous research we found that not much research has explored how prototypes were used as tools to gather requirements, Melegati et al. mentioned in their research that some organizations used prototyping as a requirements management tool but did not imply specifically how the prototype was used in the requirements engineering process. Another research study tried to use the shape of test cases. This research was conducted by [14] Found that communication between the tester and the requirements plays a crucial role. However, they don't mention how the role of the requirements used the test case to gather requirements. Therefore, we tried to incorporate the SUS as part of UAT to gather new requirements in the requirements engineering process. In this research, we want to see if using UAT, such as SUS, is adequate to gather new requirements [14]. In this research, we will use the prototyping approach. This methodology was used because the Jaklingko app itself had already been developed.

Therefore, we could use it as our prototype or MVP of the products.

MATERIALS AND METHODS

This research is Mix Method research. In order to gather the data to re-engineer the Application. Questioner for gather data Quantitative method and then interview for Qualitative method. We employed several steps. The steps can be seen in Figure 3. This step is taken from the agile scrum process. As a note, this research focused on the requirements elicitation process, not all of the agile process. This process is also used in [15]. In this research, we only focused on how to gather the requirements for redesign, not on the full development of the apps.



Source: (Research Result, 2024)
Figure 3. Research Steps

The first step in this research is to let the respondent or user try using the apps based on the task. As mentioned in the previous chapter, we used the last updated Jaklingko Apps as the MVP. The task was given to the respondent, and the time needed to finish the task will be recorded. In this step, we only give the respondents two tasks: search for a station position and find a specific route. Test case usually used as a way to evaluate the developed prototype [16].

After the respondents have done the task, the next step is for the respondents to fill out a SUS Questionnaire [11]–[13], [17]. This questionnaire collected the quantitative quality of the Application being tested. SUS is used because it needs fewer respondents around 10 peoples and can reliably collect what the user thinks about the application or system being tested [11]. Using fewer but reliable data can cut the elicitation time. This approach with agile principles where early and continuous delivery is the highest priority. We used the Indonesian translation of SUS [12] . The question asked are:

1. I think I will use this system again.
2. I find this system complicated to use.
3. I find this system easy to use.
4. I need help from others or a technician to use this system.
5. I feel that the features of this system work as they should.
6. I feel that there are many inconsistencies (lack of harmony) in this system.
7. I think other people will quickly understand how to use this system.
8. I find this system confusing.
9. I feel that there are no obstacles in using this system.
10. I need to familiarize myself before using this system.

The questionnaire will be shared with ten respondents to gather the results of the SUS questionnaire. In the next step, these respondents also became the respondents on the interview process. To calculate the SUS results. The steps are:

1. For odd number, score-1
2. For even number, 5-score
3. Sum all the results and multiply by 2.5

These SUS questionnaire results also became the basis of the interview question for qualitative data. This interview was conducted to dig into the respondent's pain points regarding their experience using the Application. The interview results are were then analyzed, identified, and compiled into the system's requirements To analyze the requirements, we used system requirement specifications to help us identify the requirements. After the requirements are gathered, we use the requirements to develop a prototype.

RESULTS AND DISCUSSION

In the first step, as mentioned before, we gave the respondents two tasks: the first was finding a specific station, and the second was finding a specific route. The results of this test can be seen in Table 1.

Table 1 Task Completion Time

Respondent	Task 1	Task 2
A	00:00:58	Not Finished
B	00:01:18	00:02:24
C	00:00:17	00:01:13
D	00:00:47	00:00:15
E	00:01:30	00:00:50
F	00:01:30	00:01:10
G	00:01:20	00:01:27
H	00:02:12	00:01:04
I	00:00:47	Not finished
J	00:01:35	00:00:57
Average	00:01:13	00:01:02
Min	00:00:17	00:00:00
Max	00:02:12	00:02:24

Source: (Research Results, 2024)

As can be seen in Table 1, the average time to finish Task 1 is one minute and thirteen seconds, and Task 2 is one minute and two seconds, which is long for a simple task. We recorded two respondents as unfinished because it took more than two minutes in the testing phase. These results show that the applications have some user interface or experience design issues. This can also be seen in the anomalous results. In Table 1 we see that two respondents can not finish. These two respondents struggle to find the best route, which also show that the design of this apps was not good enough so that the user used it intuitively. In the another hand, one respondents could easily finish the second task. Further interview revealed that the D respondents already familiar with the app because the respondent used it in their daily commute. The next step is conducting the SUS questionnaire testing. In this test, ten respondents filled out the questionnaires. The results of the SUS test can be seen in Table 2.

Table 2 SUS Questionnaire Results

Rsp	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	Pt
P1	3	2	4	1	4	2	4	4	4	5	62,5
P2	4	4	4	4	4	3	4	2	4	3	60,0
P3	4	2	4	2	5	2	4	3	4	3	72,5
P4	3	3	3	2	4	4	2	3	4	4	50,0
P5	2	3	3	2	4	2	2	4	3	4	47,5
P6	2	4	3	5	2	5	3	5	2	5	20,0
P7	3	4	2	2	2	4	1	4	2	5	27,5
P8	3	4	3	4	4	4	3	5	3	5	35,0
P9	4	2	4	3	5	2	5	3	5	5	70,0
P10	3	4	2	3	3	5	2	5	2	5	25,0
Average											47,0

Source(Research Results, 2024)

The questionnaire result shows that the Application's aggregate point is 47. Based on the SUS grading scale, this Application was graded F or Worst Imaginable[11]. These results also show that the user does not have a good experience when using the Application. This questionnaire results also show that there is connection in the results of SUS Questionnaire and the testcase results. In the Table 1 we see that some of the task was not completed. From the SUS results in Table 2 we know that in the 8th question about the confusion in using the apps. 6 respondents filled 4 and 5 which show that they are somehow confused when using the apps. Another connection in the results show in the 2nd question about the apps complexity. 5 respondents filled 4 and only 3 respondents filled 2 which shows that this apps not only confusing to used but also unnecessarily complex. From this result, we know that this Application need improvements. After that, we will interview to find the necessary improvements. From the interviews, we concluded several requirements. The requirements gathered can be seen at Table 3.

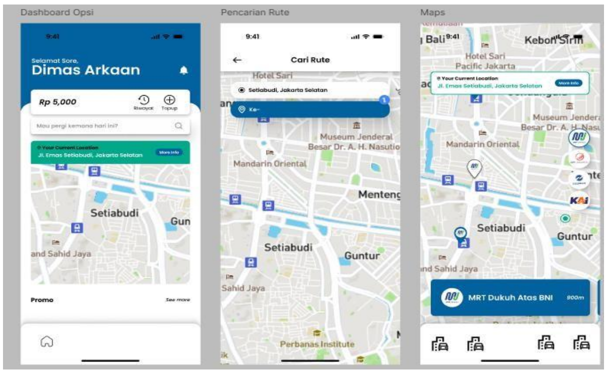
Table 3 Identified Application Requirements

Req. ID	Req. Description
Req 001	The Application could give information about the route with detailed information.
Req 002	The system could give information about the nearest Jaklingko.
Req 003	The system could give information about the nearest LRT, MRT, and BRT to the selected route.
Req 004	The Application could give the best route and be filtered by cost or time.
Req 005	The application could be the most efficient in terms of time or cost from the user's location.
Req 006	The application could provide an accurate BRT schedule.
Req 007	The application could provide a more informative map.
Req 008	The Application could give information on bus locations.
Req 09	The Application could give information about the route to the nearest bus station

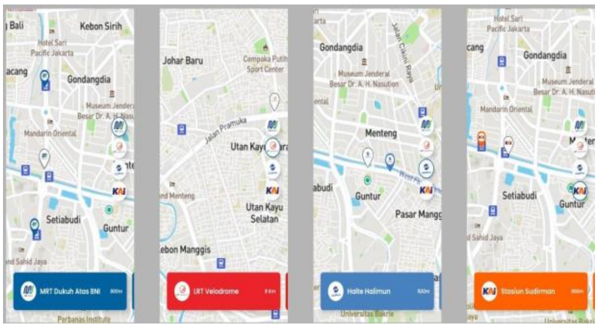
Source: (Research Results, 2024)

From the requirements of Table 3, we can see that the main problem of the application is the detailed information provided in the application. This problem is critical for an application to offer information to the user. For example, the application can provide the bus location. This application's main goal is to provide information to the user about the BRT. However, it can not give the information the user needs and frequently provides them with stressful moments. This pain point can be seen from the application review of some users, as shown in Figure 2. Therefore, some UI/UX redesign is needed based on the result gathered from our test. Figure 3 Shown the Application UI/UX redesign we developed.

In the redesign, as shown in Figure 4, we focused on giving the information needed by the user, such as the information about the route. Especially the information about the BRT location. This information is needed because the user also needs to estimate when the BRT will arrive at the station. This information is not provided by the application even though at the time of this writing; Google has already added the BRT location in their maps application. The requirements could be gathered more thoroughly if there were more test cases and all the possibilities that could be covered within the software were thoroughly covered. But, if that is the case, then it will take a lot more time to conduct and become conflicted with the Agile Manifesto, where quick and timely delivery is mandatory. Therefore, this methodology is not suited to agile requirements engineering. However, it can be used as a tool to confirm the functionality for the client or user at each sprint end.



(a)



(b)

Source:(Research Results, 2024)

Figure 4 Jaklingko Application Redesign Results (a)(b)

CONCLUSION

Based on the SUS conducted in our research,, we found that Jaklingko apps have bad UI/UX because they do not give the users proper information, such as the bus location and the best routes to the selected location. Some users also found it hard to use the apps because the UI confused them. From this pain point, we gathered the requirements and redesigned the app's UI/UX to address this pain point by adding new features and making the UI easier to navigate. In this research, we also find that the speed of requirement engineering is important in agile requirement engineering. One of the requirements engineering methods is using prototyping. Prototyping uses a product prototype or the existing application as the test subject. Using a prototype, we could grab requirements. However, the requirements gathered were not thorough and holistic because they depended on the quality of the test cases. Instead, prototyping is more suitable for requirements confirmation.

However, this research has a limitation in the number of tasks conducted by the respondents. The lack of a complete task prevented this research can grasping the application capability fully. Therefore, some of the features weren't tested and can be added in the redesign. Therefore, for future research, the number of tasks can be added so that the redesign can be done for all of the app's features. We also will conduct the SUS score for the proposed design, therefore we could compare the before and after of the application design.

REFERENCE

[1] A. O'neill, "Indonesia: Urbanisasi dari 2013 hingga 2023." [Online]. Available: https://www.statista.com/statistics/455835/urbanization-in-indonesia/

[2] A. D. Krisnanta, H. S. Hasibuan, and R. P. Tambunan, "Impact of regional infrastructure development on urbanization and environment in the northern coastal region of Central Java, Indonesia," *Sustain. Resilient Infrastruct.*, vol. 00, no. 00, pp. 1–18, 2025, doi: 10.1080/23789689.2025.2546178.

[3] M. Ipa *et al.*, "Urban-rural disparities in self-reported dengue infection: A comprehensive analysis of the 2023 Indonesian health survey [Under Review]," *Glob. Transitions J.*, vol. 8, no. 1, pp. 10–21, 2025, doi: 10.1016/j.glt.2025.08.003.

[4] N. Muhamad, "Kebanyakan Warga RI Masih Gunakan Kendaraan Pribadi Menuju Tempat Kerja," *Databoks*, no. Agustus, p. 1, 2023, [Online]. Available: https://databoks.katadata.co.id/datapublish/2023/08/24/kebanyakan-warga-ri-masih-gunakan-kendaraan-pribadi-menuju-tempat-kerja

[5] BPS Provinsi DKI Jakarta, "Rilis Berita Resmi Statistik DKI Jakarta," Jakarta, 2024. [Online]. Available: https://jakarta.bps.go.id/pressrelease/2024/03/01/1178/perkembangan-transportasi-provinsi-dki-jakarta-januari-2024.html

[6] N. N. Ferenchak and B. A. Woods, "Changes in crash types and contributing factors after bus rapid transit (BRT) infrastructure installation in Albuquerque, New Mexico," *Multimodal Transp.*, vol. 4, no. 1, p. 100192, 2025, doi: 10.1016/j.multra.2025.100192.

[7] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. de Oliveira Neto, "Requirements engineering challenges and practices in large-scale agile system development," *J. Syst. Softw.*, vol. 172, p. 110851, 2021, doi: 10.1016/j.jss.2020.110851.

[8] M. Ehn, M. Derneborg, Å. Revenäs, and A. Cicchetti, "User-centered requirements engineering to manage the fuzzy front-end of open innovation in e-health: A study on support systems for seniors' physical activity," *Int. J. Med. Inform.*, vol. 154, no. June, 2021, doi: 10.1016/j.ijmedinf.2021.104547.

[9] A. Hesham, O. E. Emam, and M. Salah, "Enhanced Framework For Big Data Requirement Elicitation," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 8, pp. 134–143, 2021, doi: 10.14569/IJACSA.2021.0120816.

[10] F. Casillo, V. Deufemia, and C. Gravino, "Detecting privacy requirements from User Stories with NLP transfer learning models," *Inf. Softw. Technol.*, vol. 146, no. July 2021, p. 106853, 2022, doi: 10.1016/j.infsof.2022.106853.

[11] Q. Khan *et al.*, "Psychometric evaluation of the System Usability Scale in the context of a childrearing app co-designed for low- and

middle-income countries,” *Digit. Heal.*, vol. 11, pp. 1–12, 2025, doi: 10.1177/20552076251335413.

[12] R. Anrahvi, N. Pratama, and S. Stevani, “Penerapan Metode System Usability Scale (SUS) dalam Mengukur Kepuasan Mahasiswa terhadap Website Direktori Akademik,” *Indones. J. Bus. Econ. Manag.*, vol. 3, no. 2, pp. 74–80, 2024, doi: 10.57152/ijbem.v3i2.2020.

[13] M. Hyzy *et al.*, “System Usability Scale Benchmarking for Digital Health Apps: Meta-analysis,” *JMIR Mhealth Uhealth*, vol. 10, no. 8, p. e37290, Aug. 2022, doi: 10.2196/37290.

[14] D. Satria, D. W. Rahma, and Q. Effendi Muftikhali, “Analyzing Engineering Approaches in Agile Software Development A Systematic Study,” *2024 3rd Int. Conf. Creat. Commun. Innov. Technol. ICCIT 2024*, pp. 1–6, 2024, doi: 10.1109/ICCIT62134.2024.10701200.

[15] E. Arce, A. Suárez-García, J. A. López-Vázquez, and M. I. Fernández-Ibáñez, “Design Sprint: Enhancing STEAM and engineering education through agile prototyping and testing ideas,” *Think. Ski. Creat.*, vol. 44, no. April, 2022, doi: 10.1016/j.tsc.2022.101039.

[16] V. Elia, G. Gnoni, and F. Tornese, “^ ĐŝĞŶĐĞ ŝƌĞĐƚ Sustainable maintenance and digital twin technology : a test case for evaluating integration potentialities,” *Procedia Comput. Sci.*, vol. 253, pp. 1840–1847, 2025, doi: 10.1016/j.procs.2025.01.246.

[17] F. D. Yudaputra, F. A. Triputra, P. W. Handayani, and N. C. Harahap, “Designing mobile-based tele dermatology for Indonesian clinic using user centred design: Quantitative and qualitative approach,” *Telemat. Informatics Reports*, vol. 16, no. November, p. 100180, 2024, doi: 10.1016/j.teler.2024.100180.