Development of a MATLAB GUI Application for Computing Reliability Indices at an Electrical Substation

Amrisal Kamal Fajri* [(1)], Erwan Tri Efendi [(1)], Bah Evan [(1)], Eri Prihatmini [(1)]
[1] *Department of Mechanical Engineering, Politeknik Negeri Semarang,* Semarang, Indonesia
Email address: ***amrisal.kamalfajri@polines.ac.id**

***Abstract*— The reliability of power distribution systems plays a crucial role in ensuring the continuous and efficient supply of electricity to consumers. This study presents the development of a MATLAB-based Graphical User Interface (GUI) application designed to compute key reliability indices, including SAIFI (System Average Interruption Frequency Index), SAIDI (System Average Interruption Duration Index), and CAIDI (Customer Average Interruption Duration Index), specifically for electrical substations. The application was developed using MATLAB's App Designer and structured to provide an intuitive and interactive user experience. Real operational data from a 150 kV substation were utilized to test and validate the application's functionality. The results demonstrate that the tool effectively simplifies the reliability analysis process, reduces calculation time, and minimizes the potential for manual errors. This application can serve as a practical support system for engineers and researchers in evaluating and improving the performance of electrical distribution systems.**

***Keywords*—** *MATLAB, GUI, SAIFI, SAIDI, CAIDI*

I. INTRODUCTION

Reliable electrical power delivery is critical for sustaining modern society and maintaining industrial, commercial, and residential operations. In the context of distribution systems, substations serve as key nodes where power transformation and routing occur; evaluation of their reliability is essential to ensure system resilience and reduce outage impacts.

Reliability performance is often quantified using standardized indices such as SAIFI (System Average Interruption Frequency Index), SAIDI (System Average Interruption Duration Index), and CAIDI (Customer Average Interruption Duration Index), which measure the frequency and duration of outages per customer over a specified period [1]. These indices form the basis for benchmarking utility performance, comparing service quality across regions, and guiding maintenance and planning decisions [2] [3].

Traditionally, reliability assessment has relied on manual processing of outage records or simulation using commercial software tools such as ETAP or proprietary analytical packages. However, these approaches can be time-consuming, error-prone, and lack transparency for non-commercial users [4]. In recent years, research has emphasized open-source and academic tools for reliability evaluation, often incorporating Monte Carlo simulation and smart-grid features; examples include tools like RELSAD and customized MATLAB code for distribution system reliability studies [5].

Graphical User Interface (GUI) development using MATLAB App Designer has facilitated more intuitive and interactive tools for power systems analysis. For instance, MATLAB GUIs have been developed to assess transient stability and system dynamics, offering clear visual feedback and usability benefits [6]. Despite these advances, there remains a gap in purpose-built GUI tools for computing reliability indices at substations using real operational data.

This research addresses that gap by developing a MATLAB GUI application that enables engineers and researchers to compute key reliability indices (SAIFI, SAIDI, CAIDI) based on outage and customer data from substations. The tool is designed to streamline data input, automate calculations, and reduce the potential for human error. To validate performance, the application is tested using actual outage records from a 150 kV substation, demonstrating efficiency gains and practical applicability.

II. METHODS

A. Research Design

This study adopts a quantitative and applied research design. It involves the systematic development of a MATLAB GUI application, followed by validation using actual substation operational data. The overall process includes data collection, computation logic design, GUI implementation, testing, and evaluation.

B. Data Collection

Outage event data—including number of interruptions, duration of each interruption, and number of affected customers—are collected from an electrical substation's historical records (for example a 150 kV substation). This dataset enables precise computation of reliability indices SAIFI, SAIDI, and CAIDI according to IEEE standard definitions [7] [8].

C. Calculation of Reliability Indices

SAIFI (System Average Interruption Frequency Index) is calculated as the total number of customer interruptions divided by the total number of customers served:

$$\text{SAIFI} = \frac{\sum \lambda i Ni}{\sum Ni} \quad (1)$$

where λi is failure count at each outage event and Ni is number of customers impacted [9] [10].

https://jurnal.polines.ac.id/index.php/eksergi

ISSN 0216-8685 (print); 2528-6889 (online)

SAIDI (System Average Interruption Duration Index) is computed as total customer interruption minutes (or hours) divided by total number of customers served:

$$SAIDI = \frac{\sum U_i N_i}{\sum N_i} \quad (2)$$

where Ui is total outage duration experienced by Ni customers [11] [12].

CAIDI (Customer Average Interruption Duration Index) is derived as the ratio of SAIDI to SAIFI:

$$SAIFI = \frac{\sum U_i N_i}{\sum \lambda_i N_i} \quad (3)$$

This index represents the average restoration time per interruption event [1].

D. *Algorithm and GUI Development*

The calculation algorithms based on the definitions above are implemented using MATLAB, and packaged in an interactive GUI constructed with App Designer. The interface allows users to upload or input outage data, automatically trigger computation, and display results in both tabular and visual formats. This approach aligns with recent academic practices in tool development for power system analysis [13] [14].

E. *Validation and Testing*

The GUI tool is validated using the real-world substation data: computed indices from the GUI are compared with manual calculations or benchmarked against known reliability performance standards (e.g. IEEE 1366-2012, national indexing norms). The correctness, usability, and time efficiency of the application are assessed.

F. *Performance Evaluation*

The tool's performance is evaluated based on:

- Accuracy, via comparison of outputs with reference or manual calculations.
- Processing time, comparing manual/commercial tool computation versus the MATLAB GUI approach.
- Error reduction, estimating incidence of user error in manual versus automated contexts.

These evaluation criteria are consistent with scholarly methods applied in reliability index assessment research [15]

III. RESULTS AND DISCUSSION

GUIDE or GUI Builder is a graphical user interface (GUI) development environment that utilizes graphical objects such as buttons, text boxes, sliders, menus, and others. Applications that employ a GUI are generally easier to learn and use, as users do not need to understand the underlying commands or how the system operates. The steps to develop a calculator application using MATLAB GUI are as follows:

1. Open the MATLAB software application.

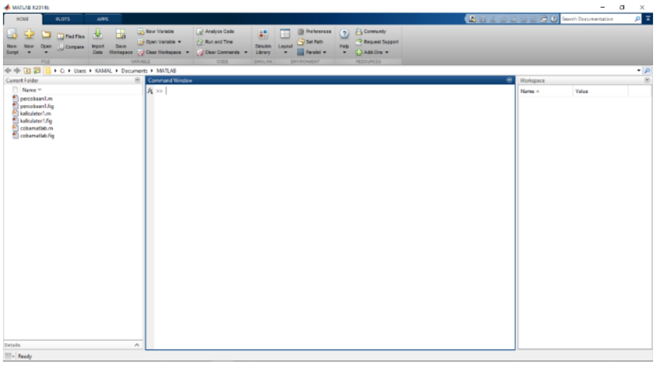

Figure 1. Initial MATLAB Interface

2. Open GUIDE in MATLAB by typing "guide" in the Command Window and pressing Enter. This will display the GUIDE Quick Start window, as shown in the figure below.

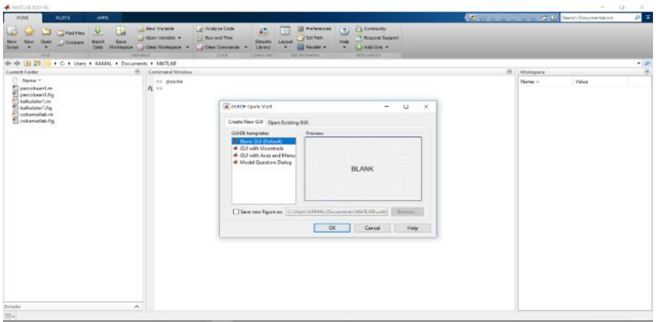

Figure 2. GUIDE Quick Start Window

Alternatively, GUIDE can be accessed by clicking on the New menu and selecting Graphical User Interface, as illustrated.

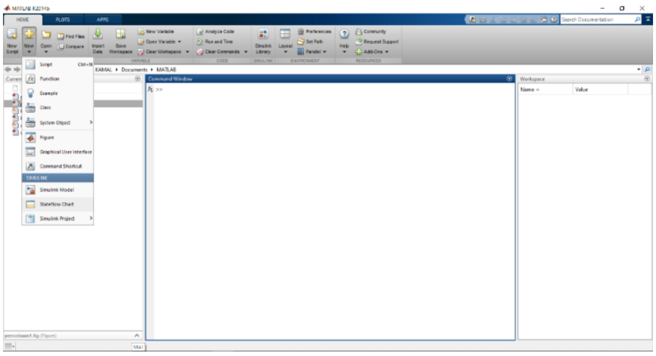

Figure 3. Starting the GUI Development

3. In the GUIDE Quick Start window, select Create New GUI >> Blank GUI (Default), then click OK.

https://jurnal.polines.ac.id/index.php/eksergi

ISSN 0216-8685 (print); 2528-6889 (online)

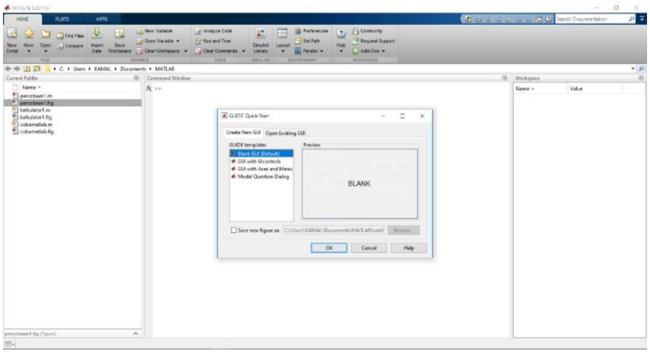

Figure 4. Creating a New GUI

4. The main GUIDE layout editor window will appear, as shown in the following figure.

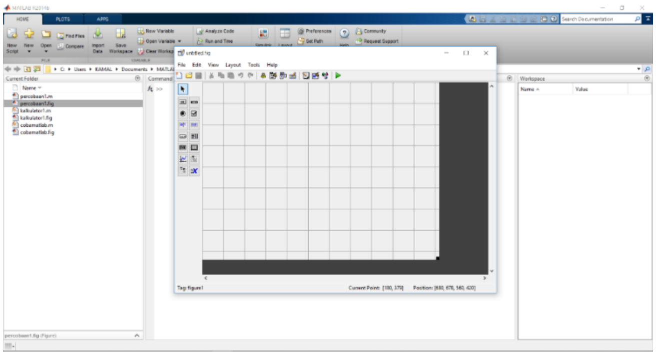

Figure 5. GUIDE Main Interface

5. To display the names of components in the palette, click File >> Preferences.

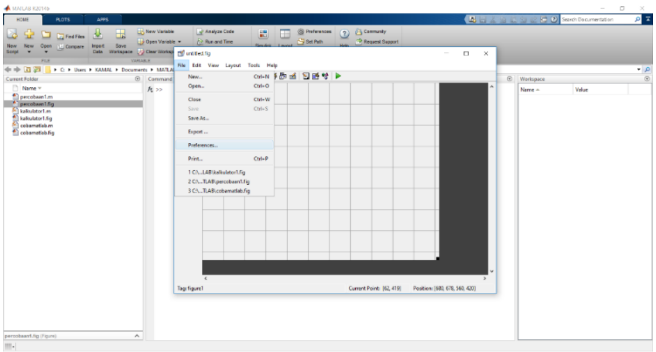

Figure 6. File Menu in the GUI

6. In the preferences window, check the option Show names in component palette, then click OK.

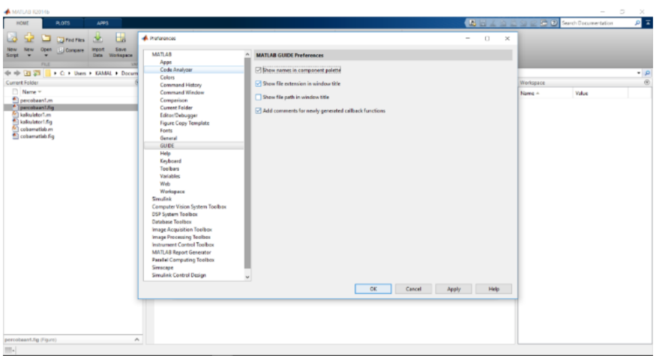

Figure 7. Preferences Menu in the GUI

7. The component palette will now display component names, as shown in the figure below.

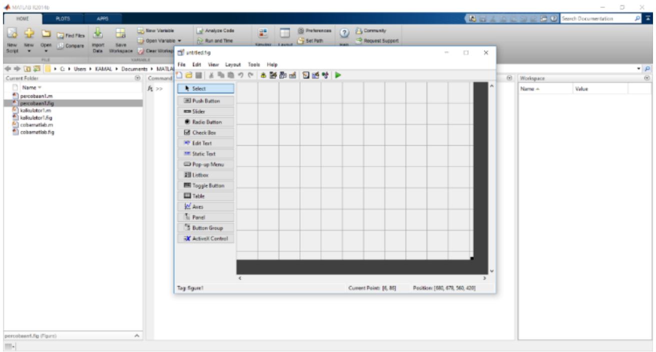

Figure 8. GUI Display Interface

8. Design the interface layout by adding the following components: 9 static text objects, 8 edit text fields, 1 pushbutton, and 1 popup menu.

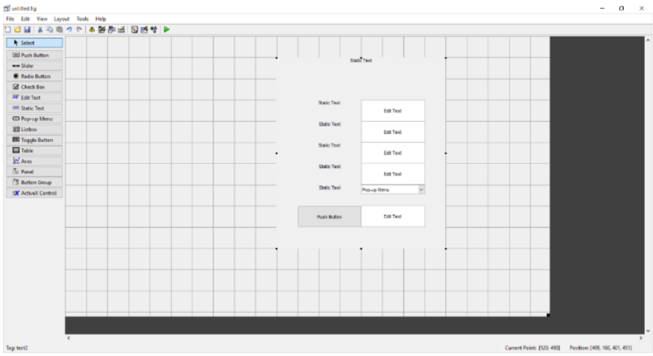

Figure 9. Figure Layout Design in GUI

9. Modify the properties of each component by double-clicking on the component and updating its properties according to the configuration table provided below.

Table 1. Component Property Inspector in MATLAB GUI

No	Nama Komponen	Property	Nilai
1	Static Text	Font Size	14
		String	Program Kalkulator MATLAB
		Tag	text1
2	Static Text	Font Size	12
		String	Input Ui
		Tag	text2
3	Static Text	Font Size	12
		String	Input Li
		Tag	text3
4	Static Text	Font Size	12
		String	Input Ni
		Tag	text4
5	Static Text	Font Size	12
		String	Input N
		Tag	text5
6	Static Text	Font Size	12
		String	SPLN
		Tag	text6
7	Static Text	Font Size	12
		String	IEEE

https://jurnal.polines.ac.id/index.php/eksergi

ISSN 0216-8685 (print); 2528-6889 (online)

		Tag	text7
		Font Size	12
8	Static Text	String	WCS dan WCC
		Tag	text8
		Font Size	12
9	Static Text	String	Masukan Rumus
		Tag	text9
		Font Size	12
10	Pushbutton	String	Hasil
		Tag	Pushbutton1
		Font Size	12
11	Popupmenu	String	SAIDI SAIFI CAIDI
		Tag	Pop up menu1
		Font Size	12
12	Edit Text	String	<kosongkan>
		Tag	edit1
		Font Size	12
13	Edit Text	String	<kosongkan>
		Tag	edit2
		Font Size	12
14	Edit Text	String	<kosongkan>
		Tag	edit3
		Font Size	12
15	Edit Text	String	<kosongkan>
		Tag	edit4
		Font Size	12
16	Edit Text	String	<kosongkan>
		Tag	edit5
		Font Size	12
17	Edit Text	String	<kosongkan>
		Tag	Edit6
		Font Size	12
18	Edit Text	String	<kosongkan>
		Tag	Edit7
		Font Size	12
19	Edit Text	String	<kosongkan>
		Tag	Edit8

10. The GUI layout will appear as shown in the figure.

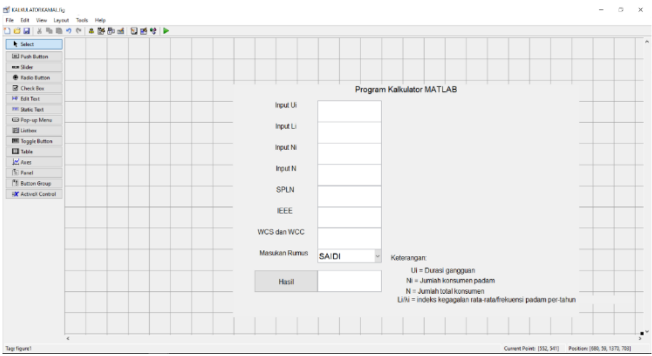

Figure 10. GUI Display After Layout Design

11. Run the GUI by clicking the Run button, then save the file with a desired filename, for example, "kalkulatorMATLAB".

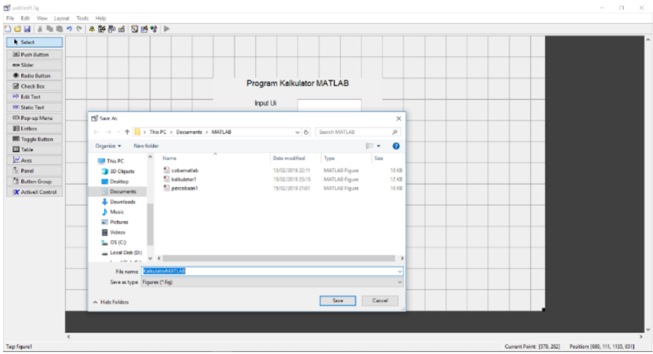

Figure 11. Saving the GUI Project

12. The resulting GUI interface will appear as shown in the following image.

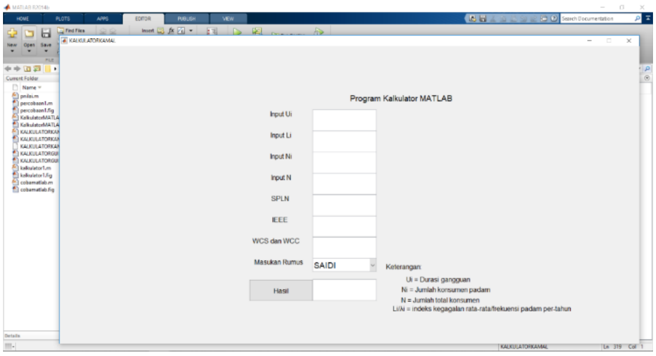

Figure 12. GUI Display During Execution

13. Insert the program script into the pushbutton1_Callback function.

```
input_ui = str2double(get(handles.edit1,'string'));
input_li = str2double(get(handles.edit2,'string'));
input_ni = str2double(get(handles.edit3,'string'));
input_n = str2double(get(handles.edit4,'string'));
spln = str2double(get(handles.edit5,'string'));
ieee = str2double(get(handles.edit6,'string'));
wcsdanwcc = str2double(get(handles.edit7,'string'));
rumus = get(handles.popupmenu1,'value');

if rumus == 1                                          → SAIDI
    hasil = input_ui * input_ni / input_n;

elseif rumus == 2                                      → SAIFI
    hasil = input_li * input_ni / input_n;

elseif rumus == 3                                      → CAIDI
   hasil = (input_ui * input_ni) / (input_li *
input_ni);
end

set(handles.edit8,'string',num2str(hasil))

if hasil<3.2,set(handles.edit5,'string','HANDAL');
else hasil>3.2,set(handles.edit5,'string','KURANG
HANDAL');
end                                                    → SPLN

if hasil<1.45,set(handles.edit6,'string','HANDAL');
else hasil>1.45,set(handles.edit6,'string','TIDAK
HANDAL');
end                                                    → IEEE

if hasil<1.66,set(handles.edit7,'string','HANDAL');
else hasil>1.66,set(handles.edit7,'string','TIDAK
HANDAL');
end                                                    → WCS dan WCC
```

Figure 13. Program Script for Pushbutton

https://jurnal.polines.ac.id/index.php/eksergi

ISSN 0216-8685 (print); 2528-6889 (online)

14. The script code will be displayed in the editor as shown in the figure.

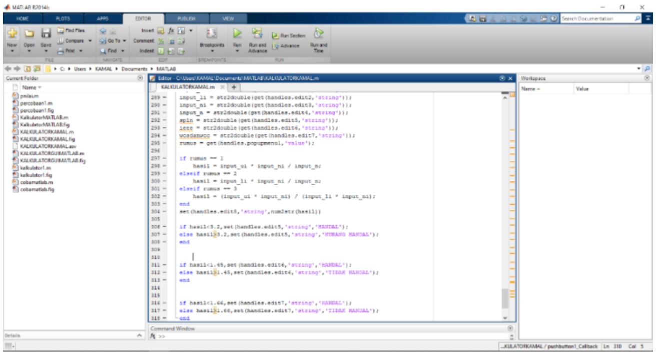

Figure 14. MATLAB Editor Interface

15. Execute the script by running it; the output interface will appear as illustrated in the figure below.

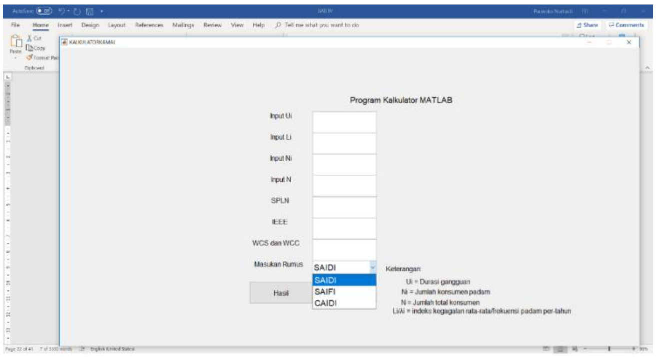

Figure 15. Executing the Script

IV. Conclusion

This study successfully developed a MATLAB-based Graphical User Interface (GUI) application for computing reliability indices—specifically SAIFI, SAIDI, and CAIDI—at an electrical substation. The application provides a user-friendly platform that enables users to input operational data and automatically calculate key reliability metrics without the need for manual computation. By integrating standardized formulas and a structured interface, the tool enhances the efficiency, accuracy, and accessibility of reliability analysis. Testing using real-world data from a 150 kV substation demonstrated that the application performs reliably and significantly reduces analysis time compared to conventional methods.

The development of this GUI tool supports engineers and researchers in monitoring and evaluating the performance of power distribution systems. It also offers potential for further integration with automated data acquisition systems or utility databases.

The author can suggest:

1. Further Development: Future work is recommended to expand the application by incorporating additional reliability indices (such as ASAI, ASUI, and ENS) and including visual trend analysis features such as charts and graphs.
2. Scalability and Integration: The tool should be adapted to handle data from multiple substations and integrated with centralized databases or SCADA systems to enable real-time reliability monitoring.
3. Validation Across Regions: The application should be tested using datasets from various substations in different regions to ensure broader applicability and robustness under diverse operational conditions.
4. User Training and Documentation: For wider adoption, comprehensive user manuals and training modules should be developed to support utility personnel and academic users in utilizing the tool effectively.

References

[1] Parol, M., Wasilewski, J., Wojtowicz, T., Arendarski, B., & Komarnicki, P. (2022). Reliability Analysis of MV Electric Distribution Networks Including Distributed Generation and ICT Infrastructure. Energies, 15(14), 5311. https://doi.org/10.3390/en15145311

[2] Prasetia, A. M., Linda Sartika, & Al Amin Hanifa Muslim. (2024). Evaluation of 20 KV Distribution System Using SAIDI and SAIFI Reliability Indices at PT PLN. Emitor: Jurnal Teknik Elektro, 24(2), 115–120. https://doi.org/10.23917/emitor.v24i2.2482

[3] Mandefro Teshome, Fsaha Mabrahtu. (2020). Reliability Assessment and Study the Effect of Substation Feeder Length on Failure Rate and Reliability Indices. American Journal of Electrical Power and Energy Systems, 9(3), 41-46. https://doi.org/10.11648/j.epes.20200903.11.

[4] Hajian-Hoseinabadi, Hamze. (2010). Reliability and component importance analysis of substation automation systems. International Journal of Electrical Power & Energy Systems - INT J ELEC POWER ENERG SYST. 49. 10.1016/j.ijepes.2010.06.012.

[5] Myhre, Stine & Fosso, Olav & Heegaard, Poul & Gjerde, O. (2021). A Tool for Reliability Assessment of Smart and Active Distribution Systems -- RELSAD. 10.48550/arXiv.2111.01195.

[6] Abboud Khoury, Muhammed Hassanain and M.Sc. Qusay Abdel Latif (2021). Transient Stability GUI with Matlab App Designer (https://www.mathworks.com/matlabcentral/fileexchange/<...>), MATLAB Central File Exchange. Retrieved April 23, 2021.

[7] Cristobal, J. R., & Santiago, R. V. M. (2025). Reliability Assessment of Power Distribution System in Freeport Area of Bataan. Engineering Proceedings, 92(1), 58. https://doi.org/10.3390/engproc2025092058

[8] Arya, R. (2016), Determination of customer perceived reliability indices for active distribution systems including omission of tolerable interruption durations employing bootstrapping. IET Gener.

https://jurnal.polines.ac.id/index.php/eksergi

ISSN 0216-8685 (print); 2528-6889 (online)

Transm. Distrib., 10: 3795-3802. https://doi.org/10.1049/iet-gtd.2016.0198

[9] Sailaja, Ch & Prasad, Polaki. (2022). Reliability evaluation of distribution system integrated with distributed generation. Bulletin of Electrical Engineering and Informatics. 11. 3166-3175. 10.11591/eei.v11i6.4534.

[10] Rausand, M., & Høyland, A. (2018). System Reliability Theory: Models, Statistical Methods, and Applications. 3rd ed. Hoboken, NJ: Wiley.

[11] Gao, Haixiang & Chen, Ying & Xu, Yin & Liu, Chen-Ching. (2016). Resilience-Oriented Critical Load Restoration Using Microgrids in Distribution Systems. IEEE Transactions on Smart Grid. 7. 1-1. 10.1109/TSG.2016.2550625.

[12] Kafle, Prakash & Bhandari, Manila & Rana, Lalit. (2022). Reliability Analysis Techniques in Distribution System: A Comprehensive Review. International Journal of Engineering and Manufacturing. 12. 11-24. 10.5815/ijem.2022.02.02.

[13] Y. Dechgummarn, P. Fuangfoo and W. Kampeerawat, "Reliability Assessment and Improvement of Electrical Distribution Systems by Using Multinomial Monte Carlo Simulations and a Component Risk Priority Index," in IEEE Access, vol. 10, pp. 111923-111935, 2022, doi: 10.1109/ACCESS.2022.3215956.

[14] Nazaruddin, & Mahalla, & Fauzi, & Maimun, & Subhan, & Abubakar, Said & Aiyub, Sayed. (2020). Reliability Analysis of 20 KV Electric Power Distribution System. IOP Conference Series: Materials Science and Engineering. 854. 012007. 10.1088/1757-899X/854/1/012007.

[15] Sukimin, Nur & Nor, Ahmad & Radzi, Nur & Eltawil, Naji. (2025). Development of a MATLAB Graphical User Interface for Predicting Bifacial Photovoltaic Panel Output. Future Energy and Environment Letters. 2. 44-62. 10.37934/feel.2.1.4462.

ISSN 0216-8685 (print); 2528-6889 (online)