DAFTAR PUSTAKA

Agalianos, K., Ponis, S. T., Aretoilaki, E., Plakas, G., dan Efthymiou,O. (2020). Discrete Event Simulation and Digital Twins: Review and Challenges for Logistics. *Journal of Procedia Manufacturing* Vo.51 (2020). https://doi.org/10.1016/j.promfg.2020.10.228.

Alam, F. M., Fowler, J. M., dan Gopalakhrishnan, M. (2012). A Discrete Event Simulation Tool For Perfomance management of Web Based Application Systems. *Journal of Simulation* 6, 21-32 (2012). https://link.springer.com/article/10.1057/jos.2011.8#Sec3.

Al-Jarrah, M. A. & Al-Masri A. M. (2023). Recent Advances in Exponential Distribution Theory and Applications. *Journal of Statistical Theory and Applications*, 19(2), 297-313. https://www.atlantis-press.com/journals/jsta/125941613.

Amazon. (2023). What is log files. https://aws.amazon.com/id/what-is/log-files/

Apache. (2024, April). Apache HTTP Server. https://httpd.apache.org/

Armitage, P., Berry, G., & Matthews, J.N.S. (2002). Statistical Methods in Medical Research. 4th ed. Blackwell Science. https://doi.org/10.1002/9780470773666

Averill, M. L. & Kelton, D. (2015). Simulation Modeling and Analysis. https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/108-Simulation-Modeling-and-Analysis-Averill-M.-Law-Edisi-5-2014.pdf.

Banks, J., John S., Nelson, B. L., & Nicol, D. M. (2015). Discrete-Event System Simulation. https://bayanbox.ir/view/2964535763283208541/DESS-JBanks-4thEd.pdf.

Calzarossa, M., & Massari, L. (2010). Analysis of Web Logs: Challenges and Findings. https://link.springer.com/chapter/10.1007/978-3-642-25575-5_19.

Capocchi, Laurent, dan Santucci, Jean F., (2021). A Web-Based Simulation of Discrete Event System of System with The Mobile Application DEVSimPymob. *Journal of ScienceDirect* Vol.13, January 2021, 100625. https://doi.org/10.1016/j.softx.2020.100625.

Capocchi, Laurent, dan Santucci, Jean F., (2022). Towards a DEVS Model Management System for Decision-Making Web Applications. *Journal of Information* 14(2), 69. https://doi.org/10.3390/info14020069.

Castillo, V. (2006). Parallel Simulations of Manufacturing Processing using Simpy, ython-Based Discrete Event Simulation Tool. *Proceedings of the 2006 ıter Simulation Conference*, 2294-2294 ɔs://doi.org/10.5555/1218112.1218537

Optimized using trial version www.balesio.com

Cohen, J., Cohen, P., West, S.G., & Aiken, L.S. (2003). Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences. Mahwah, NJ: Lawrence Erlbaum Associates. https://eli.johogo.com/Class/CCU/SEM/_Applied%20Multiple%20Regression-Correlation%20Analysis%20for%20the%20Behavioral%20Sciences_Cohen.pdf

Dejan, M. Y. M. M. (2020). Peningkatan Hasil Produksi Kertas Menggunakan Pendekatan Descrete Event Simulation (Study Kasus PT XYZ). *JISO: Journal of Industrial and Systems Optimization*, 3(2), 58-62.

Developer Mozilla. (2023, November). HTTP respons status code. https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

Eldahshan, K., Abd El-kader, A., & Ghazy, N. (2017). Achieving Stability in the Round Robin Algorithm. *International Journal of Computer Applications*. https://www.ijcaonline.org/archives/volume172/number6/eldahshan-2017-ijca-915161.pdf.

Fay, M. P. & Proschan, M. A. (2010). Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. Statist. Surv. 4 1 - 39. https://doi.org/10.1214/09-SS051

Fiona McElduff,Mario Cortina-Borja,Shun-Kai Chan, andAngie Wade. (2010). When t-tests or Wilcoxon-Mann-Whitney tests won't do. https://doi.org/10.1152/advan.00017.

Floyd, S., & Jacobson, V. (1993). Random Early Detection gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, 1(4), 397-413.

Franke, T., Ho, T., & Christie, Christina A., The Chi-Square Test. (2012). American Journal of Evaluation. https://doi.org/10.1177/1098214011426594

Gross & Haris. (2008). Fundamental Of Queveing Theory: fourth edition. John willey & sons, inc : new Jersey. https://download.e-bookshelf.de/download/0000/8064/46/L-G-0000806446-0002312173.pdf.

Haider & Dhiya. (2008). Web Log Mining Applications in Real-Life Scenarios.

Harrell, G. B. (2004). Simulation Using ProModel (2nd Edition). McGraw-Hill Companies. https://lib.ui.ac.id/detail.jsp?id=20344760.

Hostinger. (2023, April 24). Apa itu web server. https://www.hostinger.co.id/tutorial/apa-itu-web-server.

Kesar Bahasa Indonesia (KBBI). (2023). Model. https://kbbi.web.id/model.

besar Bahasa Indonesia (KBBI). (2023). Simulasi. https://kbbi.web.id/simulasi.

Optimized using
trial version
www.balesio.com

Khaire, P., Narawade, R., Badgujar, V., & Adep, Y. (2022). Web Server Analysis. *International Journal for Research in Applied Science and Engineering Technology*. https://doi.org/10.22214/ijraset.2022.41472.

Kiruthika, P. & Selvi, P. P. (2019). Web Server Log Analysis Using Web Usage Mining. *Global Journal Of Engineering Science And Researches*. https://doi-10.5281/zenodo.3174525.

Kunda, D., Chihana, S., & Sinyinda, M. (2017). Web Server Performance of Apache and Nginx: A Systematic Literature Review. Computer Engineering and Intelligent Systems.

Kurose, J.F., & Ross, K.W. (2017). Computer Networking: A Top-Down Approach. Pearson. https://www.ucg.ac.me/skladiste/blog_44233/objava_64433/fajlovi/Computer%20Networking%20_%20A%20Top%20Down%20Approach,%207th,%20converted.pdf.

Lestari, F. (2018). Simulasi Industri: Discrete Event Simulation. Cahaya Firdaus. Pekanbaru : Indonesia. https://repository.uin-suska.ac.id

Loukides, M., & Oram, A. (Eds.). (1997). Webmaster in a Nutshell. O'Reilly Media. https://www.oreilly.com/library/view/webmaster-in-a/0596003579/

Matloff, N. (2008). Introduction to Discrete-Event Simulation and the SimPy Language. UC Davis. https://heather.cs.ucdavis.edu/matloff/public_html/SimCourse/PLN/DESimIntro.pdf

Montgomery, D.C. (2012). Design and Analysis of Experiments. 8th ed. Hoboken, NJ: John Wiley & Sons. https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/181-Design-and-Analysis-of-Experiments-Douglas-C.-Montgomery-Edisi-8-2013.pdf

Morville, P., & Rosenfeld, L. (2006). Information Architecture for the World Wide Web: Designing Large-Scale Web Sites. O'Reilly Media. https://www.oreilly.com/library/view/information-architecture-for/0596527349/

Myers, Walpole. (2020). Probability and Statistics for Engineers and Scientists (9th Edition). https://spada.uns.ac.id/pluginfile.php/221008/mod_resource/content/1/ProbabilityStatistics_for_EngineersScientists(9th_Edition)_Walpole.pdf

Nayak, D. R., & Padhy, C. (2014). Revamped Round Robin Scheduling Algorithm. rnal of Engineering Computers & Applied Sciences. *Journal of gineering Computers & Applied Sciences*.

iyaroj, S. et all., Performance Evaluation of Load Balanced Web Proxies. 11). IEEE Workshops of International Conference on Advanced

Optimized using
trial version
www.balesio.com

Information Networking and Applications. https://doi.org/10.1109/WAINA.2011.129.

NGINX. (2024, March). Nginx: about. https://nginx.com/en/

Numpy. (2022). Numpy Documentation : np.random.poisson. https://numpy.org/doc/

Pahdian, M. F., Sendi, R. M., Rahmawati, R., & Dwichandra, V. D. (2021). Probabilitas Jumlah Kecelakaan Tambang di Indonesia Tahun 2019 Menggunakan Distribusi Poisson. 2(1), 7–11.

Peck, R., Olsen, C., & Devore, J.L. (2015). Introduction to Statistics and Data Analysis. Boston, MA: Cengage Learning.

Prakoso, R. D. & Asmunin. (2018). Implementasi Dan Perbandingan Performa Proxmox Dalam Virtualisasi Dengan Tiga Virtual Server. Jurnal Manajemen Informatika, 8(1).

Purnomo, R., & Putra, T. D. (2022). Case Study: Improved Round Robin Algorithm. Sinkron : Jurnal dan Penelitian Teknik Informatika. https://doi.org/10.33395/sinkron.v7i3.11530.

Rana, Rakash & Singhal R., Chi-square test and its application in hypothesis testing. (2015). Journal of the Practice of Cardiovascular Science. https://doi.org/10.4103/2395-5414.157577.

Reese, G. (2008). NGINX: The High-Performance Web Server and Reverse Proxy.

Scribbr. (2022, May 13). Poisson Distribution. https://www.scribbr.com/statistics/poisson-distribution/

Serrano, Jesus I. V., Garcia, Rodrigo E. P., dan Barron, Leopoldo E. C.,. (2021). Discrete-Event Simulation Modeling in Healthcare : A Comprehensive Review. *Journal of Enviromental Research and Public Health*. 18(22) : 12262 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8625660/.

Sheskin, D. J. (2004). Handbook of Parametric and Nonparametric Statistical Procedures. CRC Press. https://doi.org/10.1201/9781420036268

Silberschatz, A., Galvin, P.B., & Gagne, G. (2018). Operating System Concepts. John Wiley & Sons, Inc.

Statnik, A. (2023). The importance of usability in developing websites and mobile apps. *Culegere de lucrari stiintifice: Simpozion stiintific al tinerilor cercetatori*, Vol. 1. https://doi.org/10.53486/9789975359023.39

Muhammad Akbar I. F. P., Istikmal, dan Indrawan, Arif Indra. (2023). alysis of Load balancing Least Connection and Shortest Expected Delay gorithm For Web Server Using Kube-Proxy on Kubernetes. Jurnal Teknik

Optimized using
trial version
www.balesio.com

Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika Vo.11, No.2. : http://dx.doi.org/10.26760/elkomika.v11i2.439.

Tarafdar, M., & Zhang, J. (2005). Analyzing the Influence of Web Site Design Parameters on Web Site Usability. *Information Resources Management Journal*, 18, 62-80. https://www.igi-global.com/gateway/article/1281.

Tennage, Pasindu. (2019). Modeling closed System Perfomance of Server with Discreate Event Simulation. https://wso2.com/blog/research/modeling-closed-system-performance-of-a-server-with-discrete-event-simulation/.

Thau, L. (1996). Understanding Unix/Linux Programming: A Guide to Theory and Practice. Prentice Hall.

Wang, Jundi & Zhao, Kai. (2021). Performance Analysis and Optimization of Nginx-based Web Server. Journal of Physics: Conference Series. https:// doi.org/10.1088/1742-6596/1955/1/012033.

Yang, J., Di Jin, Ye Li, K. Hielscher, R. German. Modeling and simulation of performance analysis for a cluster-based Web server. (2006). https://doi.org/10.1016/j.simpat.2005.04.004

Zeigler, Bernard. (2021). DEVS-Based Building Blocks and Architectural Patterns for Intelligent Hybrid Cyberphisical System Design. *Journal of Information* 12 (12), 531. https://doi.org/10.3390/info12120531.

Optimized using
trial version
www.balesio.com

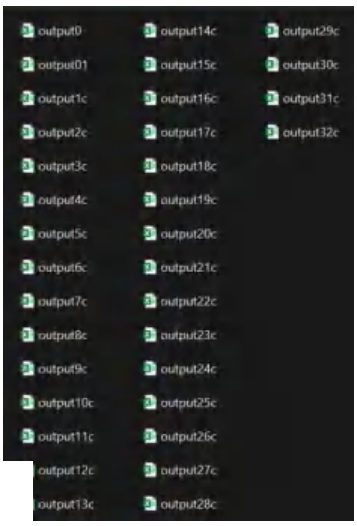
LAMPIRAN

Optimized using
trial version
www.balesio.com

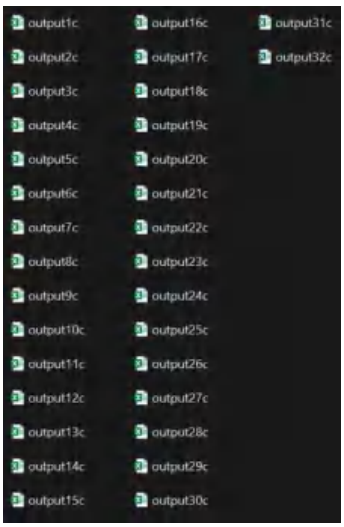
Lampiran 1. Dataset

(a) Log akses domain (bentuk .log)

(b) Log akses sub-domain (bentuk .log)



) Log akses domain (bentuk .csv)



(d) Log akses sub-domain (bentuk .csv)

Optimized using
trial version
www.balesio.com

Lampiran 2. Pra-pemrosesan data (*data preprocessing*)

```python
#mengubah .log ke .csv
import csv

# Nama file input dan output
input_file = 'access.log'
output_file = 'access.csv'

# Membuka file teks untuk dibaca
with open('D:\\Tes Data Penelitian\\accesslog\\access.log', 'r') as txt_file:
    # Membaca setiap baris dari file teks
    lines = txt_file.readlines()

# Mengonversi data teks menjadi list of lists
data = [line.strip().split() for line in lines]

# Menentukan header file CSV (berdasarkan asumsi bahwa setiap kolom dipisahkan oleh spasi)
csv_header = ['ip_address', 'timestamp', 'requests_method', 'path', 'status_code', 'response_size', 'referrer', 'user_agent']

# Menulis data ke dalam file CSV
with open('access.csv', 'w', newline='') as csv_file:
    csv_writer = csv.writer(csv_file)
    csv_writer.writerow(csv_header)
    csv_writer.writerows(data)
```

(a) Mengubah .log menjadi .csv

```python
#cleaning data
import pandas as pd

# Membaca file CSV
csv_file_path = 'D:\\Tes Data Penelitian\\accesslog\\access.csv'
df = pd.read_csv(csv_file_path)

# Menampilkan informasi umum tentang data
print("Info sebelum data cleaning:")
print(df.info())

# Menghapus duplikat (jika ada)
df = df.drop_duplicates()

# Menangani nilai-nilai yang hilang
df = df.dropna()

# Mengonversi kolom timestamp menjadi tipe data datetime
df['timestamp'] = pd.to_datetime(df['timestamp'], format='%d/%b/%Y:%H:%M:%S %z')

# Menyimpan kembali ke file CSV setelah data cleaning
cleaned_csv_path = 'D:\\Tes Data Penelitian\\accesslog\\tes\\output.csv'
df.to_csv(cleaned_csv_path, index=False)

# Menampilkan informasi setelah data cleaning
print("\nInfo setelah data cleaning:")
print(df.info())
```

(b) Pembersihan data

```python
# Menghitung interval arrival permintaan
import pandas as pd

# Membaca file CSV yang telah dibersihkan
csv_path = 'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output1c.csv'
# csv_path = 'D:\\Tes Data Penelitian\\accesslog\\access.csv'
df = pd.read_csv(csv_path, low_memory=False)

# Mengkonversi kolom timestamp menjadi tipe data datetime
df['timestamp'] = pd.to_datetime(df['timestamp'], format='%Y-%m-%d %H:%M:%S%z')

# Mengurutkan DataFrame berdasarkan timestamp
df = df.sort_values(by='timestamp')

# Menghitung interval kedatangan
df['interval_arrival'] = df['timestamp'].diff()

# Menyimpan DataFrame yang telah diupdate ke file CSV
updated_csv_path = 'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output0.csv'
df.to_csv(updated_csv_path, index=False)
```

(c) Menghitung interval arrival permintaan

Optimized using
trial version
www.balesio.com

```python
# Menghitung total data keseluruhan
import pandas as pd

# Daftar alamat file CSV yang akan dibaca
csv_files = [
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output1c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output2c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output3c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output4c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output5c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output6c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output7c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output8c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output9c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output10c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output11c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output12c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output13c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output14c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output15c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output16c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output17c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output18c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output19c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output20c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output21c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output22c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output23c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output24c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output25c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output26c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output27c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output28c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output29c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output30c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output31c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output32c.csv'
]

# Inisialisasi list kosong untuk menyimpan DataFrames dari setiap file
dfs = []

# Loop melalui setiap alamat file CSV, membaca mereka, dan menambahkannya ke list
for file in csv_files:
    dfs.append(pd.read_csv(file))

# Menggabungkan semua DataFrames menjadi satu DataFrame
combined_df = pd.concat(dfs, ignore_index=True)

# Menampilkan jumlah baris dan kolom dari DataFrame gabungan
print("\nDimensi Data Gabungan:")
print(combined_df.shape)
```

(d) Menghitung total data keseluruhan

```python
# Melihat banyak data dalam file CSV
import pandas as pd

# Membaca file CSV
csv_file_path = 'D:\\Tes Data Penelitian\\log_webhosting\\output1c.csv'
df = pd.read_csv(csv_file_path)

# Menampilkan jumlah baris dan kolom
print("\nDimensi Data:")
print(df.shape)

# Menampilkan informasi umum tentang data
print("\nInformasi Data:")
print(df.info())
```

(e) Melihat banyak data dalam setiap file

Optimized using
trial version
www.balesio.com

Lampiran 3. Mencari distribusi data

```
# Mencari rata-rata request per jam domain
import pandas as pd
import matplotlib.pyplot as plt

# Daftar nama file CSV
file_paths = ['D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output1c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output2c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output3c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output4c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output5c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output6c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output7c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output8c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output9c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output10c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output11c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output12c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output13c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output14c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output15c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output16c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output17c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output18c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output19c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output20c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output21c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output22c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output23c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output24c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output25c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output26c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output27c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output28c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output29c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output30c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output31c.csv',
]

# Membaca data dari setiap file dan menghitung jumlah permintaan per jam
requests_per_hour = {}
for i, file_path in enumerate(file_paths, start=1):
    df = pd.read_csv(file_path)
    df['timestamp'] = pd.to_datetime(df['timestamp'], errors='coerce')
    df = df.dropna(subset=['timestamp'])
    df['hour'] = df['timestamp'].dt.hour
    requests_per_hour[f'File_{i}'] = df.groupby('hour').size()

# Gabungkan data dari semua file
combined_requests = pd.concat(requests_per_hour.values(), axis=1).sum(axis=1)

# Hitung total permintaan gabungan
total_requests = combined_requests.sum()

# Hitung rata-rata permintaan per jam dari data gabungan
average_requests_per_hour = combined_requests / len(requests_per_hour)

# Hitung persentase permintaan per jam dari total permintaan
percentage_requests_per_hour = (combined_requests / total_requests) * 100
# Visualisasikan rata-rata permintaan per jam dalam bentuk histogram
plt.figure(figsize=(10, 6))
bars = plt.bar(average_requests_per_hour.index, average_requests_per_hour, color='skyblue', width=0.6)

plt.title('Rata-rata Permintaan per Jam Domain\n 11 November 2023 - 11 Desember 2023')
plt.xlabel('Jam')
plt.ylabel('Rata-rata Permintaan')
plt.grid(axis='y')

# Menambahkan keterangan jam di bawah setiap batang histogram
plt.gca().xaxis.set_ticks(range(len(average_requests_per_hour)))
plt.gca().xaxis.set_ticklabels(average_requests_per_hour.index)

# Menambahkan label di atas atau di dalam batang histogram
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, height + 5, str(int(height)), ha='center', va='bottom')

plt.tight_layout()
plt.show()

# Visualisasikan persentase permintaan per jam dalam bentuk histogram
plt.figure(figsize=(10, 6))
bars = plt.bar(percentage_requests_per_hour.index, percentage_requests_per_hour, color='salmon', width=0.6)

plt.title('Persentase Permintaan per Jam Domain\n 11 November 2023 - 11 Desember 2023')
plt.xlabel('Jam')
plt.ylabel('Persentase Permintaan')
plt.grid(axis='y')

# Menambahkan keterangan jam di bawah setiap batang histogram
plt.gca().xaxis.set_ticks(range(len(percentage_requests_per_hour)))
plt.gca().xaxis.set_ticklabels(percentage_requests_per_hour.index)

# Menambahkan label di atas batang histogram
for bar in bars:
    ar.get_height()
    ar.get_x() + bar.get_width() / 2, height, f'{height:.2f}%', ha='center', va='bottom')

    ut()


    a-rata permintaan per jam untuk setiap file
    requests_per_hour_file in requests_per_hour.items():
    quests_file = requests_per_hour_file.mean()
    ta-rata Permintaan per Jam untuk {file_name}: {average_requests_file:.2f}')
```

(a) Mencari distribusi jumlah permintaan

Optimized using
trial version
www.balesio.com

```python
# Melihat distribusi interval
import pandas as pd

# Daftar alamat file CSV yang akan dibaca
file_paths = [
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output1c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output2c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output3c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output4c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output5c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output6c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output7c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output8c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output9c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output10c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output11c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output12c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output13c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output14c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output15c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output16c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output17c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output18c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output19c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output20c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output21c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output22c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output23c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output24c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output25c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output26c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output27c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output28c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output29c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output30c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output31c.csv'
]

# Membaca data dari setiap file dan menggabungkannya
dfs = []
for file_path in file_paths:
    df = pd.read_csv(file_path, low_memory=False)
    dfs.append(df)

combined_df = pd.concat(dfs)

# Menghitung jumlah total data
total_data = len(combined_df)

# Menghitung jumlah permintaan dan persentasenya untuk setiap interval kedatangan
interval_counts = combined_df['interval_arrival'].value_counts()

# Menghitung rata-rata permintaan
average_requests = total_data / len(interval_counts)

print("Interval Arrival\tJumlah Permintaan\tPersentase")
print("-------------------------------------------------")
for interval, count in interval_counts.items():
    percentage = (count / total_data) * 100
    print(f"{interval}\t\t\t{count}\t\t\t{percentage:.2f}%")
```

(b) Mencari distribusi interval arrival

Optimized using
trial version
www.balesio.com

```python
import pandas as pd
import os
import matplotlib.pyplot as plt

# Daftar nama file CSV
file_paths = ['D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output1c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output2c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output3c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output4c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output5c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output6c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output7c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output8c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output9c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output10c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output11c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output12c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output13c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output14c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output15c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output16c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output17c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output18c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output19c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output20c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output21c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output22c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output23c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output24c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output25c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output26c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output27c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output28c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output29c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output30c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output31c.csv',
]
# Membaca data dari setiap file dan menggabungkannya menjadi satu DataFrame
dfs = []
for file_path in file_paths:
    df = pd.read_csv(file_path)
    dfs.append(df)

combined_df = pd.concat(dfs, ignore_index=True)

# Ubah kolom 'timestamp' menjadi tipe data waktu
combined_df['timestamp'] = pd.to_datetime(combined_df['timestamp'])

# Menghitung jumlah keseluruhan status code dalam gabungan semua file
total_status_codes = len(combined_df['status_code'])

# Menghitung jumlah masing-masing status code
status_code_counts = combined_df['status_code'].value_counts()

# Menghitung persentase masing-masing status code
status_code_percentages = {}
for status_code, count in status_code_counts.items():
    percentage = (count / total_status_codes) * 100
    status_code_percentages[status_code] = percentage

# Filter status code yang valid
valid_status_codes = {str(code): count for code, count in status_code_counts.items() if isinstance(code, int)}

# Plot pie chart
plt.figure(figsize=(8, 8))
plt.pie(valid_status_codes.values(), startangle=140)
plt.title('Status Code Domain\n 11 November 2023 - 11 Desember 2023')
plt.axis('equal')  # Ensure the pie is drawn as a circle

# Add legend
legend_texts = [f"{status_code}: {count} ({percentage:.2f}%)"
                for status_code, count, percentage in zip(valid_status_codes.keys(),
                                                          valid_status_codes.values(),
                                                          status_code_percentages.values())]
plt.legend(legend_texts, loc='lower left')
plt.show()
```

(c) Mencari distribusi kode status (*status code*)

Optimized using
trial version
www.balesio.com

```
import pandas as pd
import os
import matplotlib.pyplot as plt

# Daftar nama file CSV
file_paths = ['D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output1c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output2c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output3c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output4c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output5c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output6c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output7c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output8c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output9c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output10c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output11c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output12c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output13c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output14c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output15c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output16c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output17c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output18c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output19c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output20c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output21c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output22c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output23c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output24c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output25c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output26c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output27c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output28c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output29c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output30c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output31c.csv',
]

# Membaca data dari setiap file dan menggabungkannya menjadi satu DataFrame
dfs = []
for file_path in file_paths:
    df = pd.read_csv(file_path)
    dfs.append(df)

combined_df = pd.concat(dfs, ignore_index=True)

# Filter hanya method request yang valid (GET, POST, HEAD, OPTIONS)
valid_request_methods = ['GET', 'POST', 'HEAD', 'OPTIONS']
combined_df = combined_df[combined_df['request_method'].isin(valid_request_methods)]

# Menghitung jumlah keseluruhan request method yang valid
total_request_method = len(combined_df)

# Menghitung jumlah masing-masing request method
request_method_counts = combined_df['request_method'].value_counts()

# Menghitung persentase masing-masing request method
request_method_percentages = {}
for request_method, count in request_method_counts.items():
    percentage = (count / total_request_method) * 100
    request_method_percentages[request_method] = percentage

# Plot pie chart untuk request method
plt.figure(figsize=(8, 8))
plt.pie(request_method_counts, startangle=140)
plt.title('Request Method Domain\n 11 November 2023 - 11 Desember 2023')
plt.axis('equal')  # Ensure the pie is drawn as a circle

# Add legend
legend_texts = [f"{method}: {count} ({percentage:.2f}%)"
                for method, count, percentage in zip(request_method_counts.index,
                                                     request_method_counts.values,
                                                     request_method_percentages.values())]
plt.legend(legend_texts, loc='lower left')

# Show the plot
plt.show()
```

(d) Mencari distribusi metode permintaan (*requests method*)

Optimized using
trial version
www.balesio.com

```
# Mencari distribusi referrer
import pandas as pd
import matplotlib.pyplot as plt

def extract_domain(url):
    # Ekstrak domain dengan kemungkinan 'http://', 'https://', dan 'www'
    if not pd.isna(url):
        start = url.find("://www.")
        if start == -1:
            start = url.find("://")
        if start != -1:
            start += 3
        else:
            start = 0
        end = url.find("/", start)
        if end == -1:
            return url[start:]
        else:
            return url[start:end]
    return np.nan

# Daftar nama file CSV
file_paths = ['D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output1c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output2c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output3c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output4c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output5c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output6c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output7c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output8c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output9c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output10c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output11c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output12c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output13c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output14c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output15c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output16c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output17c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output18c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output19c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output20c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output21c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output22c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output23c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output24c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output25c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output26c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output27c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output28c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output29c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output30c.csv',
    'D:\\Tes Data Penelitian\\accesslog\\tes\\sudah cleaning\\output31c.csv']

# Membaca data dari setiap file dan menggabungkannya menjadi satu DataFrame
dfs = []
for file_path in file_paths:
    df = pd.read_csv(file_path)
    dfs.append(df)

combined_df = pd.concat(dfs, ignore_index=True)

# Menerapkan fungsi ekstraksi domain ke kolom 'referrer'
combined_df['domain'] = combined_df['referrer'].apply(extract_domain)

# Hitung kemunculan setiap domain
domain_counts = combined_df['domain'].value_counts()
domain_percentages = domain_counts / domain_counts.sum() * 100  # Calculate percentages

# Buat label untuk diagram lingkaran
labels = [f'{domain} - {count} - {percentage:.2f}%' for domain, count, percentage in zip(domain_counts.index, domain_counts, do

plt.figure(figsize=(10, 8))
plt.pie(domain_counts, labels=None, autopct='%1.1f%%', startangle=90)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.legend(labels, title="Domains", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))

plt.title('Referrer Domain/n 11 November 2023 - 11 Desember 2023')
plt.show()
```

(e) Mencari distribusi referrer

Optimized using
trial version
www.balesio.com

Lampiran 4. Uji parameter

```
[ ]  hourly_requests = [
                5400, 6400, 5500, 4700, 4900, 4900, 5000, 4500,
                3700, 3300, 3700, 3500, 3500, 3100, 2400, 1700,
                1100, 800, 650, 400, 1000, 1200, 2500, 4200]
     num_requests = np.random.poisson(hourly_requests)
     print(f"num requests = {num_requests}")

     num requests = [5358 6497 5468 4721 4930 4791 5005 4458 3773 3352 3736 3514 3540 3159
      2402 1689 1076  845  592  428 1020 1197 2546 4104]
```

(a) Uji parameter jumlah permintaan per jam

```
[ ]  import random
     from scipy.stats import expon

     zero_interval_percentage = 0.8
     lambda_param = 1.0

     if random.random() < zero_interval_percentage:
         interval_time = 0
     else:
         interval_time = expon.rvs(scale=1/lambda_param)

     print(f"interval time : {interval_time}")

     interval time : 0
```

(b) Uji parameter interval permintaan

```
import numpy as np

# Dictionary status_codes
status_codes = {
    200: 53.90,
    404: 38.18,
    500: 3.64,
    503: 2.62,
    301: 1.07,
    403: 0.15,
    499: 0.09,
    206: 0.08,
    302: 0.06,
    400: 0.05,
    201: 0.04
}
# Mendefinisikan status code sebagai daftar
codes = list(status_codes.keys())

# Mendefinisikan probabilitas dari masing-masing status code
total_probability = sum(status_codes.values())
probabilities = [v / total_probability for v in status_codes.values()]
# Uji coba parameter menggunakan np.random.choice
selected_code = np.random.choice(codes, p=probabilities)
print("Status code yang terpilih:", selected_code)

Status code yang terpilih: 200
```

(c) Uji parameter kode status

```
import numpy as np

# Dictionary request_method
request_methods = {
            'GET': 95.18,
            'POST': 4.76,
            'HEAD': 0,04,
            'OPTIONS': 0.01
        }
# Mendefinisikan request method sebagai daftar
codes = list(request_methods.keys())

# Mendefinisikan probabilitas dari masing-masing request method
total_probability = sum(request_methods.values())
probabilities = [v / total_probability for v in request_methods.values()]

# Uji coba parameter menggunakan np.random.choice
selected_code = np.random.choice(codes, p=probabilities)
print("Method request yang dipilih:", selected_code)

Method request yang dipilih: GET
```

(d) Uji parameter metode permintaan

Optimized using
trial version
www.balesio.com

```python
import numpy as np

# Dictionary referrers
referrers = {
    'https://www.unhas.ac.id': 80,
    'http://www.unhas.ac.id': 10,
    'https://id.images.search.yahoo.com': 3,
    'https://images.search.yahoo.com': 2.5,
    'https://www.google.com': 1.5,
    'http://hotspot.rsgm': 1,
    'http://fkuhs3.tes': 1,
    'https://www.bing.com': 1
}

# Mendefinisikan referer sebagai daftar
referers = list(referrers.keys())

# Mendefinisikan proporsi probabilitas
total_visits = sum(referrers.values())
probabilities = [v / total_visits for v in referrers.values()]

# Uji coba parameter menggunakan np.random.choice
selected_referer = np.random.choice(referers, p=probabilities)

print("Referer yang terpilih:", selected_referer)
```

```
Referer yang terpilih: https://www.unhas.ac.id
```

(e) Uji parameter referrer

Tautan *Source code* lengkap simulator web server:

https://github.com/Rschnrlhdyt/simulator-webserver

Tautan buku panduan penggunaan:

https://drive.google.com/drive/folders/1C9cX-aVBWU4THsESy9bMk4O-b8RJ-Aud?usp=sharing

Optimized using
trial version
www.balesio.com

Lampiran 5. Uji hipotesis

```python
#uji hipotesis rata-rata permintaan per jam (domain)
import numpy as np
from scipy import stats

# Data contoh
data_real = np.array([1735, 2230, 2475, 2235, 1970, 2055, 1990, 2110, 1910, 1625, 1555,
           1720, 1660, 1685, 1545, 1315, 955, 745, 575, 535, 430, 650, 755, 1220])
data_sim = np.array([1735, 2230, 2471, 2221, 1977, 2076, 2001, 2112, 1895, 1620, 1559,
           1734, 1657, 1693, 1551, 1313, 953, 745, 567, 535, 437, 645, 750, 1228])

# Tes normalitas
print("Normality Test Real:", stats.shapiro(data_real))
print("Normality Test Sim:", stats.shapiro(data_sim))

# Tes homogenitas varians
print("Levene's Test for Equal Variance:", stats.levene(data_real, data_sim))

# Uji T dua sampel independen
t_stat, p_val = stats.ttest_ind(data_real, data_sim, equal_var=True)
print("T-test Statistic:", t_stat, "P-value:", p_val)
```

```
Normality Test Real: ShapiroResult(statistic=0.9312734603881836, pvalue=0.10409270226955414)
Normality Test Sim: ShapiroResult(statistic=0.928758442401886, pvalue=0.0914541631937027)
Levene's Test for Equal Variance: LeveneResult(statistic=0.00011871929700206075, pvalue=0.9913536695541751)
T-test Statistic: -0.00586198899976751 P-value: 0.9953481852760175
```

(a) Uji hipotesis jumlah rata-rata permintaan

```python
#uji hipotesis status code (domain)
import numpy as np
import scipy.stats as stats

# Membuat array numpy dari data
data_real = np.array([53.90, 38.18, 3.64, 2.62, 1.07, 0.15, 0.09, 0.08, 0.06, 0.05, 0.04])
data_simulasi = np.array([53.95, 38.25, 3.65, 2.62, 1.06, 0.15, 0.09, 0.08, 0.06, 0.05, 0.04])

# Membuat tabel kontingensi
table = np.array([data_real, data_simulasi])

chi2, p, dof, expected = stats.chi2_contingency(table)

print("Chi-square Statistic:", chi2)
print("p-value:", p)
```

```
Chi-square Statistic: 7.591387362484875e-05
p-value: 1.0
```

(b) Uji hipotesis kode status

```python
#uji hipotesis referrer (domain)
import numpy as np
import scipy.stats as stats

# Membuat array numpy dari data
data_real = np.array([80.00, 10.00, 3.00, 2.50, 1.50, 1.00, 1.00, 1.00])
data_simulasi = np.array([79.99, 10.06, 2.99, 2.48, 1.48, 0.99, 1.00, 1.00])

# Membuat tabel kontingensi
table = np.array([data_real, data_simulasi])

chi2, p, dof, expected = stats.chi2_contingency(table)

print("Chi-square Statistic:", chi2)
print("p-value:", p)
```

```
Chi-square Statistic: 0.00046108185053050926
p-value: 0.999999999999984
```

(c) Uji hipotesis referrer

Optimized using
trial version
www.balesio.com

```
[ ]  #uji hipotesis method requests (domain)
     import numpy as np
     import scipy.stats as stats

     # Membuat array numpy dari data
     data_real = np.array([95.18, 4.76, 0.04, 0.01])
     data_simulasi = np.array([95.18, 4.77, 0.03, 0.01])

     # Membuat tabel kontingensi
     table = np.array([data_real, data_simulasi])

     chi2, p, dof, expected = stats.chi2_contingency(table)

     print("Chi-square Statistic:", chi2)
     print("p-value:", p)

     Chi-square Statistic: 0.0014390646080047966
     p-value: 0.9999854871816409
```

(d) Uji hipotesis metode permintaan

Optimized using
trial version
www.balesio.com

Lampiran 6. Daftar hadir dan berita acara seminar hasil

KEMENTERIAN PENDIDIKAN , KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus FakultasTeknikUnhas, Jl. PorosMalino, Gowa
http://eng.unhas.ac.id/informatika, Email : informatika@unhas.ac.id

DAFTAR HADIR SEMINAR HASIL

Nama/Stambuk : 1. Rischa Nurul Hidayat D121201068

Judul Skipsi/T.A : **"Rancang Bangun Simulator Web Server Guna Menguji Kinerja Server Menggunakan Discrete Event Simulation "**

Hari/Tanggal : Jumat, 17 Mei 2024

Jam : 14.30 Wita – Selesai

Tempat : Ruang Lab. CBS Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Mukarramah Yusuf,B.Sc.,M.Sc.,Ph.D	1 .....
	Pembimbing II	2. Dr. Eng. Ady Wahyudi Paundu,ST.,M.T	2 .....
II.	Anggota Penguji	3. Dr. Eng. Zulkifli Tahir,ST.,M.Sc	3 .....
		4. Iqra Aswad,ST.,M.T	4 .....

PANITIA UJIAN

Ketua,

Mukarramah Yusuf,B.Sc.,M.Sc.,Ph.D

Sekretaris,

Dr. Eng. Ady Wahyudi Paundu,ST.,M.T

Optimized using
trial version
www.balesio.com

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
http://eng.unhas.ac.id/informatika, Email : informatika@unhas.ac.id

Nomor : 583/UN4.7.7/TD.06/2024
Lamp : -
Hal : Penerbitan Surat Penugasan Panitia/Penguji Seminar Hasil Strata Satu (S1)

Kepada Yth :

Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Universitas Hasanuddin

Di-

Gowa

Dengan hormat,
Berdasarkan Persetujuan Pembimbing Mahasiswa, Bersama ini diusulkan susunan Panitia/Penguji Seminar Hasil Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik tersebut di bawah ini :

Nama / Stambuk : Rischa Nurul Hidayati D121201068
Judul TA : Rancang Bangun Simulator Web Server Guna Menguji Kinerja Server Menggunakan Discrete Event Simulation

Dengan ini kami sampaikan Susunan Panitia Seminar Hasil Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I/ Ketua : 1. Mukarramah Yusuf, B.Sc., M.Sc., Ph.D
Pembimbing II / Sekretaris : 2. Dr.Eng. Ady Wahydi Paundu, ST., M.T.
Anggota : 3. Dr.Eng. Zulkifli Tahir, ST., M.Sc.
4. Iqra Aswad, ST., M.T.

Untuk dapat diterbitkan surat penugasannya

Demikian penyampaian kami, atas perhatian dan kerjasamanya diucapkan terima kasih.

Gowa, 14 Mei 2024
Ketua Departemen Tek.Informatika,

Prof. Dr. Ir. Indrabayu, ST, MT., M.Bus.Sys., IPM, ASEAN.Eng
Nip.19750716 200212 1 004

Tembusan :
1. Arsip

Jumat, 17 Mei 2024
Jam : 14.30
Lab : CBS

Optimized using
trial version
www.balesio.com

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
http://eng.unhas.ac.id, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 10379/UN4.7.1/TD.06/2024

Dari : Dekan Fakultas Teknik Universitas Hasanuddin

Kepada : Mereka yang tercantum namanya dibawah ini

Isi :

1. Bahwa merujuk kepada Peraturan Rektor Universitas Hasanuddin **Nomor : 29/UN4.1/2023 tentang Penyelenggaraan Program Sarjana Universitas Hasanuddin** , dengan ini menugaskan Saudara sebagai PENGUJI/PANITIA SEMINAR HASIL Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

   Pembimbing I/ Ketua : 1. Mukarramah Yusuf, B.Sc., M.Sc., Ph.D
   Pembimbing II / Sekretaris : 2. Dr.Eng. Ady Wahydi Paundu, ST., M.T.
   Anggota : 3. Dr.Eng. Zulkifli Tahir, ST., M.Sc.
   4. Iqra Aswad, ST., M.T.

   Untuk menguji bagi mahasiswa tersebut dibawah ini :

   Nama/NIM : Rischa Nurul Hidayati D121201068
   Program Studi : Teknik Informatika
   Judul thesis/Skrips : Rancang Bangun Simulator Web Server Guna Menguji Kinerja Server Menggunakan Discrete Event Simulation

2. Waktu seminar ditetapkan oleh Panitia Seminar Hasil Program Strata Satu (S1)
3. Agar Surat Penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasa ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudia hari terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa
Pada tanggal 14 Mei 2024
a.n. Dekan,
Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas

Dr. Amil Ahmad Ilham, ST., M.IT
NIP. 197310101998021001

Tembusan :
1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Mahasiswa yang bersangkutan

Balai Sertifikasi Elektronik
• Dokumen ini telah ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan BSrE
• UU ITE No 11 Tahun 2008 Pasal 5 Ayat 1

Optimized using
trial version
www.balesio.com

Lampiran 7. Daftar hadir dan berita acara ujian skripsi

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus FakultasTeknikUnhas, Jl. PorosMalino, Gowa
http://eng.unhas.ac.id/informatika, Email : informatika@unhas.ac.id

DAFTAR HADIR UJIAN SKRIPSI MAHASISWA
FAKULTAS TEKNIK UNHAS

Nama/Stambuk : 1. Rischa Nurul Hidayati D121201068

Judul Skipsi/T.A : **"Rancang Bangun Simulator Web Server Guna Menguji Kinerja Server Menggunakan Discrete Event Simulation "**

Hari/Tanggal : Rabu, 12 Juni 2024
Jam : **09.30** Wita – Selesai
Tempat : Ruang Lab. CBS Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Mukarramah Yusuf,B.Sc.,M.Sc.,Ph.D	1. .....
	Pembimbing II	2. Dr. Eng. Ady Wahyudi Paundu,ST.,M.T	2. .....
II.	Anggota Penguji	3.Dr. Eng. Zulkifli Tahir,ST.,M.Sc	3 .....
		4. Iqra Aswad,ST.,M.T	4 .....

PANITIA UJIAN

Ketua,

Mukarramah Yusuf,B.Sc.,M.Sc.,Ph.D

Sekretaris,

Dr. Eng. Ady Wahyudi Paundu,ST.,M.T

Optimized using
trial version
www.balesio.com

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan
☎ +62811 4420 909, E-mail: teknik@unhas.ac.id , https://eng.unhas.ac.id

Gowa, 10 Juni 2024

Nomor : 759/UN4.7.7.1/TD.06/2024
Lamp : -
Hal : Usulan Susunan Panitia/Penguji Ujian Sarjana

Yth. : Bapak Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas
Di
Gowa

Dalam rangka penyelesaian studi pada Departemen Teknik Informatika Fakultas Teknik Unhas, bersama ini kami usulkan susunan Panitia/Penguji Ujian Sarjana Program Strata Satu (S1) bagi mahasiswa Departemen Teknik Informatika Fakultas Teknik Uniersitas Hasanuddin atas nama :

Pembimbing I / Ketua	: 1. Mukarramah Yusuf, B.Sc., M.Sc., Ph.D.
Pembimbing II / Sekretaris	: 2. Dr.Eng. Ady Wahyudi Paundu, ST., M.T.
Anggota	: 3. Dr.Eng. Ir. Zulkifli Tahir, ST., M.Sc.
	4. Iqra Aswad, ST., M.T.

Untuk Bertugas sebagai Penguji/ Penanggap Ujian Sarjana bagi Mahasiswa :

Nama : Rischa Nurul Hidayati
Stambuk : D121 20 1068

Dengan Judul Skripsi

" Rancang Bangun Simulator Web Server Guna Menguji Kinerja Server Menggunakan Discrete Event Simulation "

Pada :
Hari/Tanggal : Rabu, 12 Juni 2024
Jam : 09.30 Wita - Selesai
Tempat : Ruang Sidang Lab. CBS

Demikian penyampaian kami, atas perhatiannya diucapkan terimah kasih.

Ketua Departemen Tek.Informatika,

Prof. Dr. Ir. Indrabayu.,ST, MT, M.Bus.Sys., IPM, ASEAN.Eng
Nip.197507016 200212 1 004

Tembusan :
1. Arsip

Optimized using
trial version
www.balesio.com

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
Jalan Poros Malino Km. 6 Bontomarannu, Gowa, 92171, Sulawesi Selatan
☎ +62811 4420 909, E-mail: teknik@unhas.ac.id , https://eng.unhas.ac.id

SURAT PENUGASAN

No. 13232/UN4.7.1/TD.06/2024

Dari : Dekan Fakultas Teknik Universitas Hasanuddin.

Kepada. : Mereka yang tercantum namanya di bawah ini.

Isi : 1. Bahwa merujuk kepada Peraturan Rektor Universitas Hasanuddin **Nomor : 29/UN4.1/2023 tentang Penyelenggaraan Program Sarjana Universitas Hasanuddin**, dengan ini menugaskan Saudara sebagai **PENGUJI/PANITIA UJIAN SARJANA** Program Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua	: 1. Mukarramah Yusuf, B.Sc., M.Sc., Ph.D.
Pembimbing II / Sekretaris	: 2. Dr.Eng. Ady Wahyudi Paundu, ST., M.T.
Anggota	: 3. Dr.Eng. Ir. Zulkifli Tahir, ST., M.Sc.
	4. Iqra Aswad, ST., M.T.

untuk menguji bagi mahasiswa tersebut di bawah ini :

Nama/NIM	: Rischa Nurul Hidayati D121 20 1068
Program Studi	: Teknik Informatika
Judul Thesis/Skripsi	: " Rancang Bangun Simulator Web Server Guna Menguji Kinerja Server Menggunakan Discrete Event Simulation "

2. Waktu Ujian ditetapkan oleh Panitia Ujian Sarjana Program Strata Satu (S1).
3. Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Ujian Sarjana tersebut, dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,
Pada tanggal 10 Juni 2024
a.n. Dekan
Wakil Dekan Bidang Akademik dan Kemahasiswaan
Fakultas Teknik Unhas

Dr. Amil Ahmad Ilham, ST., M.IT
NIP.197310101998021001

Tembusan :
1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Kasubag. Umum dan Perlengkapan FT-UH

Optimized using
trial version
www.balesio.com

Lampiran 8. Lembar perbaikan skripsi

LEMBAR PERBAIKAN SKRIPSI

"RANCANG BANGUN SIMULATOR WEB SERVER GUNA MENGUJI KINERJA SERVER MENGGUNAKAN *DISCRETE EVENT SIMULATION*"

OLEH:

RISCHA NURUL HIDAYATI
D121 20 1068

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 12 Juni 2024.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Mukarramah Yusuf, B.Sc., M.Sc., Ph.D.	
Sekretaris	Dr. Eng. Ady W. Paundu, S. T., M. T.	
Anggota	Iqra Aswad, S.T., M.T.	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Mukarramah Yusuf, B.Sc., M.Sc., Ph.D.	
II	Dr. Eng. Ady W. Paundu, S. T., M. T.	

Optimized using
trial version
www.balesio.com