Journal of Social Research
P-ISSN: 2827-9832
E-ISSN: 2828-335x
http://ijsr.internationaljournallabs.com/index.php/ijsr

Implementation of the Critical Path Method (CPM) in Web Applications for Project Scheduling with Python Programming

Meliska Eviany Raranta, Fabian J Manoppo, Grace Yoyce Malingkas
Program Pascasarjana, Universitas Sam Ratulangi, Indonesia
Email: meliskararanta03@gmail.com, fabian_jm@unsrat.ac.id, grace@unsrat.ac.id

ABSTRACT

The construction industry faces complex challenges involving multiple phases, numerous stakeholders, and intricate interdependencies among activities. The Critical Path Method (CPM) is a proven technique for planning, organizing, and monitoring digital project scheduling activities. This research aims to develop a web-based application that implements CPM for project scheduling using the Python programming language. The study utilizes data from the *Pancang Tiowor* project by *PT. Natural Indococonut Organik*, with a contract value of *IDR.* 5,145,000,000.00 and a 90-day execution period in 2024. The methodology comprises a literature review, requirements analysis, and CPM algorithm implementation, all integrated with an interactive interface built using the Streamlit framework. The application automatically computes Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), and Slack values, while also providing network diagram visualizations. Validation results indicate that the web application delivers accurate calculations, consistent with both manual computations and Microsoft Project outputs. The identified critical path duration is 73 days, compared to the planned 90-day project duration, which provides a 17-day buffer for non-critical tasks. The application effectively offers an efficient, accessible, and user-friendly solution for digital project scheduling management.

Keywords: critical path method, project scheduling, web application, python programming, construction management

INTRODUCTION

The construction industry faces many complex problems because it involves multiple stages, the number of parties involved, and the dependencies between very complex activities (Alaloul et al., 2016). Time management and scheduling of activities are important components in construction project management so that work can be completed on time, within budget, and of the highest quality (Ervianto, 2005). The Critical Path Method (CPM) is an efficient approach in planning, organizing, and monitoring construction project activities (Dwiretnani & Kurnia, 2018).

CPM is applied to identify critical paths in a construction project, i.e. a series of activities with the longest duration where a delay in one of the activities will delay the entire project (Bishnoi, 2018). This method helps project managers optimize turnaround time by ensuring activities on critical paths are completed on time; while analyzing the relationships between activities and estimating the time it will take to complete the overall project (Kerzner, 2017).

While CPM is very useful, its application in construction projects often faces technical challenges, especially in processing complex data. Many project organizers still use manual methods to calculate CPMs or software that is less flexible and difficult to use (Wasito & Syaikhudin, 2020). In today's digital age, the need for technological tools to manage project schedules more efficiently is increasingly important (Polii et al., 2017).

Python is a highly flexible programming language that comes with a variety of libraries and frameworks that support web application development (Rawat, 2020). Using frameworks such as Streamlit, it is possible to develop web applications that integrate CPM methods to automatically calculate and visualize critical paths on construction projects (Saabith et al., 2021).

The Critical Path Method (CPM) is a project management technique used to determine the minimum duration required to complete a project (Morris, 2021). CPM identifies critical activities that do not have a float and forms the longest path in the project network (Iwawo et al., 2016). This method was first developed in the 1950s by the DuPont Company to manage large projects (Pattiraja et al., 2024).

The main components in CPM include:

1. Early Start (ES): The earliest the activity can start
2. Early Finish (EF): The earliest the activity can be completed
3. Late Start (LS): The last time an activity can start without delaying the project
4. Late Finish (LF): The last time an activity can be completed without delaying the project
5. Float/Slack: Activity time allowance (LS - ES or LF - EF)

The development of information technology has encouraged the use of web applications in various fields, including construction project management (Pamungkas et al., 2020). Web applications offer advantages in terms of accessibility, ease of maintenance, and better team collaboration (Ramadhan & Putra, 2022). Frameworks such as Streamlit allow for the easy development of interactive web applications using Python (Kansha, 2023).

Python has become one of the most popular programming languages for application development due to its simple syntax, comprehensive library, and large community (Tan et al., 2023). In the context of project management, Python provides various libraries such as NetworkX for graph analysis, Pandas for data manipulation, and Matplotlib for visualization.

The novelty of this research lies in the integration of the Critical Path Method (CPM) into a web-based application using Python and Streamlit. While previous studies, such as those by Kusumadewi & Hartati (2023), have focused on CPM implementation using conventional software like Microsoft Project and Primavera, these tools have limitations in flexibility and ease of access. Additionally, Wijayanto & Setiawan (2022) attempted to apply CPM using Python, but without utilizing the interactive and efficient potential of the Streamlit framework. This research aims to bridge that gap by developing a more flexible, user-friendly, and accessible web-based CPM application, optimizing project scheduling in a simpler, faster, and more efficient way. The integration of Python-Streamlit promises a significant improvement over traditional tools by offering real-time updates and interactive interfaces for project managers, thus enhancing the scheduling process in construction projects.

This research aims to develop a web-based application that applies the Critical Path Method (CPM) method to simplify the process of planning and managing project schedules using the Python programming language. This application is expected to speed up planning, save time, and make it easier to monitor construction projects.

RESEARCH METHOD

This study uses data from the Pancang Tiowor project carried out by PT. Natural Indococonut Organic with a contract value of IDR 5,145,000,000.00 and an implementation period of 90 calendar days in the 2024 fiscal year. The location of the project is in Kupa Kupa Sel., Tobelo Sel. District, North Halmahera Regency, North Maluku.

This research uses a software engineering approach with the following stages:

1. Literature Study: Gather references on CPM, web applications, and Python programming
2. Needs Analysis: Identify the features needed in the application
3. System Design: Designing application architecture and user interface
4. Implementation: Developing applications using Python and Streamlit
5. Testing and Validation: Comparing results with manual calculations and Microsoft Project

The data used includes:
1. Activity sequence number
2. Activity name
3. Activity notation (A, B, C, etc.)
4. Duration of implementation (days)
5. Predecessor activities

Tools and technology used were:
1. Python: The main programming language
2. Streamlit: Framework for web application development
3. NetworkX: A library for graph analysis
4. Pandas: A library for data manipulation
5. Matplotlib: A library for visualization
6. Visual Studio Code: Integrated Development Environment (IDE)
7. GitHub: Version control and repository
8. Streamlit Cloud: Platform deployment

The implementation of the CPM algorithm in the application includes:
1. Forward Pass: Calculating ES and EF
2. ES(activity) = max(EF(predecessor))
3. EF(activity) = ES(activity) + Duration
4. Backward Pass: Calculating LS and LF
5. LF(activity) = min(LS(successor))
6. LS(activity) = LF(activity) - Duration
7. Float Calculation:
8. Float = LS - ES = LF - EF
9. Critical Path Identification: Activities with Float = 0

RESULTS AND DISCUSSION

Project Activity Data

The data used in the application of CPM includes:
1. No.: Sequential number for each activity.
2. Activity: The name of the task or work in the project.
3. Notation: A unique identification for each activity (e.g., A, B, C).
4. Duration (Days): The time required to complete the activity, measured in days.
5. Predecessor Activities: Activities that must be completed before this activity can begin.

Table 1. Activity Data

No.	Activity	Code	Duration (Days)	Predecessor Activities
1	Earth Work	A	5	-
2	Reinforcement Steel	B	1	A
3	Formwork	C	1	A
4	Concrete Work	D	2	B
5	Structural Steel	E	2	D
6	Sundries	F	15	B, C, E
7	Embankment preparation	G	1	F
8	Excavation	H	1	F
9	Stone foundation installation	I	1	F
10	Install wooden columns and beams	J	1	I
11	Sand and gravel, t≤15 cm	K	1	J

No.	Activity	Code	Duration (Days)	Predecessor Activities
12	Plastic sheet	L	1	J
13	Concrete floor casting, t=10 cm	M	1	K, L
14	Roof truss installation	N	1	J, M
15	Wall installation, t≤1 m	O	1	N
16	Roof installation	P	1	N
17	Cladding wall installation	Q	1	O, P
18	Door installation	R	1	Q
19	Electrical installation	S	1	Q
20	Sand and gravel, t>15 cm	T	1	R, S
21	Foundation pedestal excavation	U	1	T
22	Footing concrete	V	1	T
23	Concrete floor casting	W	1	U, V
24	Wiremesh M8, 1 layer	X	1	W
25	Lightweight steel columns and beams	Y	1	W
26	Battens 40.45 @1m and bracing	Z	1	Y
27	Wind ties	AA	1	Z
28	Zinc roofing	AB	1	Y,Z,AA
29	Wall, t≤1 m	AC	1	AB
30	Cladding	AD	1	AB
31	Electrical and lighting installation	AE	2	AC,AD,AB
32	Embankment preparation	AF	2	AE
33	Excavation	AG	1	AE
34	Stone foundation installation	AH	1	AF,AG,AE
35	Install wooden columns, beams, and trusses	AI	2	AH
36	Roof installation	AJ	1	AH
37	Fascia board installation	AK	1	AI,AJ
38	Plastic sheet	AL	1	AK
39	Floor ceramic tiles 40x40	AM	2	AK
40	Lean concrete floor casting, t=8 cm	AN	1	AK
41	Wall installation, t≤1 m	AO	1	AN
42	Ctrc wall installation, t=9 mm	AP	1	AO
43	Bathroom (1x2 m, door, tiles, faucet, squat toilet)	AQ	1	AP
44	Double wooden door installation	AR	1	AO,AP
45	Door + frame installation (0.9x2 m)	AS	1	AR
46	Window + frame installation (60x100)	AT	1	AR
47	Window + frame installation (60x100x2)	AU	1	AR
48	Window + frame installation (60x100x3)	AV	1	AR
49	Electrical installation incl. panel & conduit	AW	1	AO,AP,AV
50	Lights incl. switches	AX	1	AW
51	Electrical outlets	AY	1	AW
52	Plumbing installation incl. floor drain and septic tank	AZ	1	AW
53	Concrete kitchen counter incl. sink	BA	1	AZ
54	Floor formwork	BB	1	BA
55	Wiremesh M8, double layer	BC	1	BB
56	Foundation pedestal excavation	BD	1	BB,BC
57	3" galvanized pipe posts	BE	1	BD
58	Pedestal concrete	BF	1	BD
59	Flat foundation concrete casting, t=20 cm	BG	1	BE,BF,BD
60	Spandek roof + hollow beam	BH	1	BG
61	Weighing house	BI	1	BG
62	Electrical installation	BJ	1	BH,BI,BG
63	Office gypsum ceiling, 9 mm	BK	1	BJ
64	Coconut chute L=60 cm, plate t=4 mm	BL	1	BK
65	Security post, size 2x2 m	BM	1	BK
66	Garuda PLUN house, size 11x11 m	BN	1	BL,BM,BK
67	Perimeter fence with piles + frame and foundation	BO	4	BN
68	Excavation + sorting warehouse frame, L=2 m	BP	1	BO
69	Pen + east and west side ring of sorting warehouse	BQ	1	BP

No.	Activity	Code	Duration (Days)	Predecessor Activities
70	Railing + walkway	BR	1	BQ

Critical Path Calculation

The determination of the critical path using Microsoft Project is carried out by entering activity data, duration, and dependencies between activities. Once the data is complete, Microsoft Project automatically calculates the critical path and displays it visually, usually with a red line, indicating the sequence of activities with no time slack.

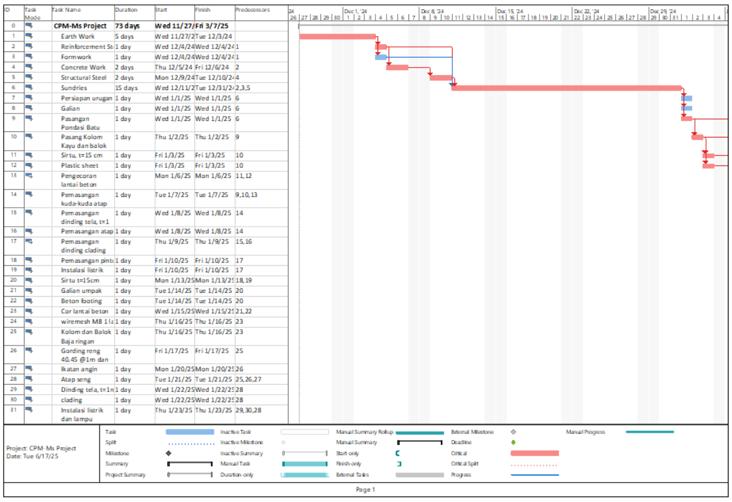


Figure 1. Microsoft Project Data Preview

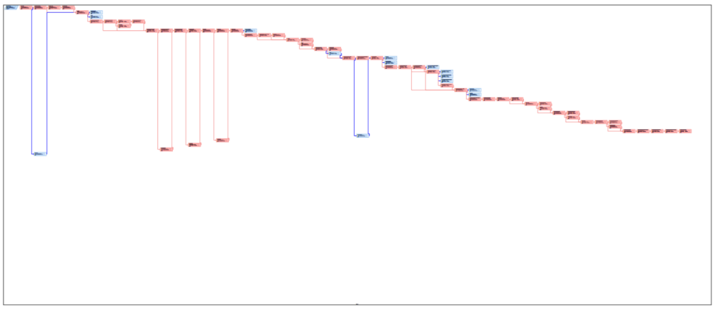


Figure 2. CPM Network Diagram in MS Project

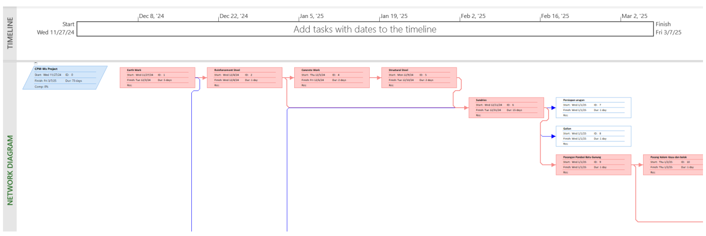


Figure 3. Segment of the CPM Network Diagram in MS Project

Using the Critical Path Method (CPM) through Microsoft Project, the result shows that the duration of the project's critical path is 73 working days. This duration is the result of a crashing process applied to the initial project timeline, which was initially planned for 90 working days. The acceleration process was carried out while considering the interrelationships between activities and the available resources, resulting in a more efficient critical path.

This result indicates that the minimum time required to complete all project activities without delay on the critical path is 73 working days. Compared to the original plan, this gives a time reserve (float) of 17 working days, which can be utilized for non-critical activities without affecting the overall project schedule.

CPM Calculation Table

After calculating and visualizing the critical path, the results are presented in a table. This table shows the Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), and Slack Time for each activity in the project. The calculation of this table is done using Microsoft Excel, where data on durations and activity dependencies are input first. Then, a forward pass is performed to obtain the values for ES and EF, followed by a backward pass to get the values for LS and LF. The difference between the early and late start and finish times is used to calculate Slack Time, which indicates the time flexibility of an activity without affecting the overall project schedule. This table serves as the basis for analysis before comparing and validating the results with the web application.

Table 2. Calculation of ES, EF, LF, and Slack

Activity node	Preceding Activities	Duration	ES	EF	LS	LF	Slack	Critical?
A	-	5	0	5	0	5	0	Yes
B	A	1	5	6	5	6	0	Yes
C	A	1	5	6	9	10	4	No
D	B	2	6	8	6	8	0	Yes
E	D	2	8	10	8	10	0	Yes
F	B,C,E	15	10	25	10	25	0	Yes
G	F	1	25	26	72	73	47	No
H	F	1	25	26	72	73	47	No
I	F	1	25	26	25	26	0	Yes
J	I	1	26	27	26	27	0	Yes
K	J	1	27	28	27	28	0	Yes
L	J	1	27	28	27	28	0	Yes
M	K,L	1	28	29	28	29	0	Yes
N	I,J,M	1	29	30	29	30	0	Yes
O	N	1	30	31	30	31	0	Yes
P	N	1	30	31	30	31	0	Yes
Q	O,P	1	31	32	31	32	0	Yes
R	Q	1	32	33	32	33	0	Yes
S	Q	1	32	33	32	33	0	Yes
T	R,S	1	33	34	33	34	0	Yes
U	T	1	34	35	34	35	0	Yes
V	T	1	34	35	34	35	0	Yes
W	U,V	1	35	36	35	36	0	Yes
X	W	1	36	37	72	73	36	No
Y	W	1	36	37	36	37	0	Yes
Z	Y	1	37	38	37	38	0	Yes
AA	Z	1	38	39	38	39	0	Yes
AB	Y,Z,AA	1	39	40	39	40	0	Yes
AC	AB	1	40	41	40	41	0	Yes
AD	AB	1	40	41	40	41	0	Yes
AE	AC,AD,AB	1	41	42	41	42	0	Yes
AF	AE	2	42	44	42	44	0	Yes
AG	AE	1	42	43	43	44	1	No
AH	AF,AG,AE	1	44	45	44	45	0	Yes
AI	AH	2	45	47	45	47	0	Yes
AJ	AH	1	45	46	46	47	1	No
AK	AI,AJ	1	47	48	47	48	0	Yes
AL	AK	1	48	49	72	73	24	No
AM	AK	2	48	50	71	73	23	No
AN	AK	1	48	49	48	49	0	Yes

Activity node	Preceding Activities	Duration	ES	EF	LS	LF	Slack	Critical?
AO	AN	1	49	50	49	50	0	Yes
AP	AO	1	50	51	50	51	0	Yes
AQ	AP	1	51	52	72	73	21	No
AR	AO,AP	1	51	52	51	52	0	Yes
AS	AR	1	52	53	72	73	20	No
AT	AR	1	52	53	72	73	20	No
AU	AR	1	52	53	72	73	20	No
AV	AR	1	52	53	52	53	0	Yes
AW	AO,AP,AV	1	53	54	53	54	0	Yes
AX	AW	1	54	55	72	73	18	No
AY	AW	1	54	55	72	73	18	No
AZ	AW	1	54	55	54	55	0	Yes
BA	AZ	1	55	56	55	56	0	Yes
BB	BA	1	56	57	56	57	0	Yes
BC	BB	1	57	58	57	58	0	Yes
BD	BB,BC	1	58	59	58	59	0	Yes
BE	BD	1	59	60	59	60	0	Yes
BF	BD	1	59	60	59	60	0	Yes
BG	BE,BF,BD	1	60	61	60	61	0	Yes
BH	BG	1	61	62	61	62	0	Yes
BI	BG	1	61	62	61	62	0	Yes
BJ	BH,BI,BG	1	62	63	62	63	0	Yes
BK	BJ	1	63	64	63	64	0	Yes
BL	BK	1	64	65	64	65	0	Yes
BM	BK	1	64	65	64	65	0	Yes
BN	BL,BM,BK	1	65	66	65	66	0	Yes
BO	BN	4	66	70	66	70	0	Yes
BP	BO	1	70	71	70	71	0	Yes
BQ	BP	1	71	72	71	72	0	Yes
BR	BQ	1	72	73	72	73	0	Yes

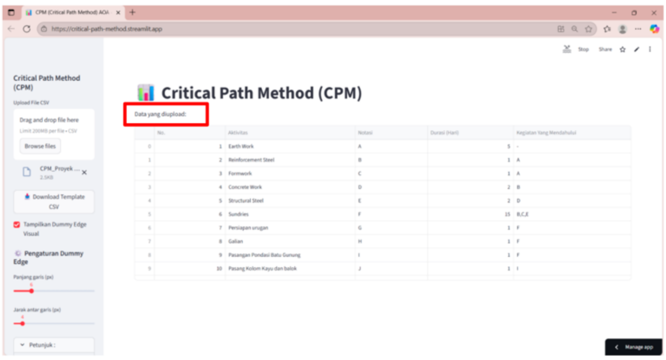


Figure 4. Data Upload CSV

Critical Path Visualization (CPM Chart)

Displays the project activity network in the form of a graph:

CPM Calculation Results

Based on the calculations using the developed web application, the following results were obtained:

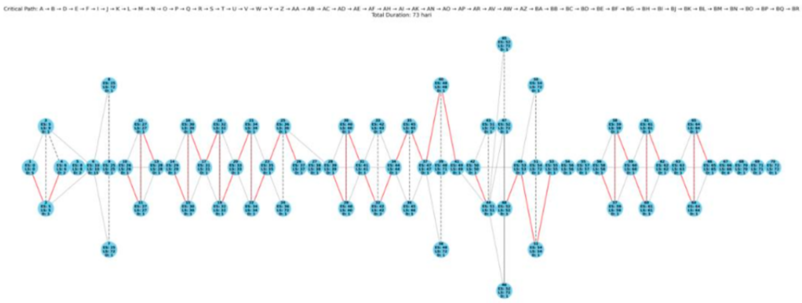
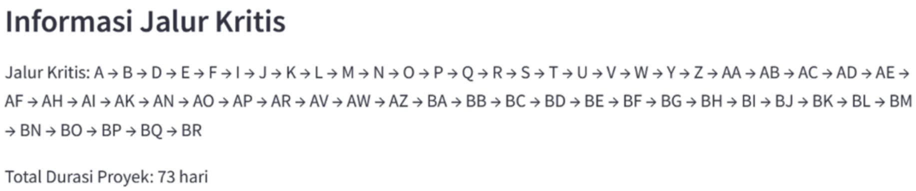


Figure 5. Critical Path Chart

Figure 6. Critical Path Segment

This figure shows:
Node: Displays information such as Activity Number, ES (Early Start), LS (Late Start), and Duration.
Edge: The connecting lines between activities that indicate the project flow.
Critical Path: Activities that determine the project duration, highlighted in red.

Critical Path Information
Displays the main results of the calculation:
Critical Path: The sequence of activities without slack (time flexibility).
Total Project Duration: Calculated from the critical path. This information is crucial for project planning and monitoring.



Informasi Jalur Kritis

Jalur Kritis: A → B → D → E → F → I → J → K → L → M → N → O → P → Q → R → S → T → U → V → W → Y → Z → AA → AB → AC → AD → AE → AF → AH → AI → AK → AN → AO → AP → AR → AV → AW → AZ → BA → BB → BC → BD → BE → BF → BG → BH → BI → BJ → BK → BL → BM → BN → BO → BP → BQ → BR

Total Durasi Proyek: 73 hari

Figure 7. Critical Path Information

Contains numerical results for each activity:
1) ES (Early Start)
2) EF (Early Finish)
3) LS (Late Start)
4) LF (Late Finish)
5) Slack (**Slack Time Table**)

This table helps project managers identify which activities are flexible and which ones must not be delayed.

Web Application Implementation

Application Features

The developed CPM web application has the following features:
1. Main Dashboard: Displays app overview and usage instructions
2. Upload Data: A feature for uploading CSV files containing activity data
3. Download Template: Provides format-appropriate CSV templates
4. Graph Visualization: Displays a network of critical path diagrams
5. Results Table: Displays CPM calculation results in table form
6. Critical Path Information: Displays activities that are in critical paths
7. Data Validation: Checking the consistency and completeness of input data

Application Architecture

The application is developed with a simple architecture consisting of:
1. Frontend: Streamlit framework for user interface
2. Backend: Python scripts for business logic and CPM calculations
3. Data Processing: Pandas for CSV data manipulation
4. Visualization: NetworkX and Matplotlib for graph visualization
5. Deployment: Streamlit Cloud for application hosting

CPM Algorithm Implementation

The implementation of the CPM algorithm in the application includes several main functions:

```
def calculate_cpm(date):
    # Create a directed graph
    G = nx. DiGraph()
    # Adding nodes and edges
    for _, row in data.iterrows():
        G.add_node(row['Notation'], duration=row['Duration'])
    # Forward pass calculation
    for node in nx.topological_sort(G):
        # Calculate ES and EF
    # Backward pass calculation
    for node in reversed(list(nx.topological_sort(G))):
        # Calculate LS and LF
    # Identify critical paths
    critical_path = [node for node in G.nodes() if G.nodes[node]['float'] == 0]
    return results
```

Validation and Testing

Comparison with Microsoft Project

Validation is done by comparing the results of the web application with the calculations using Microsoft Project. The results show 100% consistency for:
1) Total project duration (73 days)
2) Identify critical paths
3) Values of ES, EF, LS, LF for each activity
4) Float/slack calculation

Functionality Testing

App testing includes:
1. Test Upload Data: Successfully read CSV files in various formats
2. Test Calculation: Accurate results for various project scenarios
3. Test Visualization: Graph network diagram displays correctly
4. Test Validation: The system can detect errors in input data
5. Test Responsiveness: Apps run smoothly across multiple devices

Deployment and Accessibility

The application was successfully deployed using Streamlit Cloud and can be accessed via URL: https://critical-path-method.streamlit.app/. The advantages of this deployment include:
1. Global Accessibility: Accessible from anywhere with an internet connection
2. Multi-Platform: Compatible with desktops, tablets, and smartphones
3. Auto-Update: Automated code changes are deployed
4. Zero Maintenance: No server management required

Application Performance Analysis

Advantages of the App
1. User-Friendly: Simple and intuitive interface
2. Accurate: The calculation results are in accordance with CPM standards
3. Efficient: Fast and automated calculation process
4. Portable: Accessible from a variety of devices
5. Cost-Effective: Free and open source

Application Limitations
1. Internet Dependence: Requires an internet connection for access
2. Project Size: Optimal for projects of medium complexity
3. Limited Features: Doesn't support resource allocation and cost analysis yet
4. Data Format: Only supports CSV format input

Practical Implications

The results of the study show that the Pancang Tiowor project has a critical path of 73 days compared to the 90-day implementation plan. This indicates:
1. Buffer Time: There is a reserve time of 17 days (18.9% of the total duration)
2. Flexibility: Non-critical activities have room for customization
3. Risk Management: Buffer time can anticipate unexpected delays
4. Resource Optimization: Resources can be allocated more effectively

CONCLUSION

Based on the results, this study concludes that the web-based application developed using Python and Streamlit successfully implements the Critical Path Method (CPM) for project scheduling in the construction sector. The application is capable of automatically calculating and visualizing essential scheduling parameters, including Early Start, Early Finish, Late Start, Late Finish, Slack, and critical paths, as well as generating network diagrams that enhance project monitoring and decision-making efficiency. The validation using the Pancang Tiowor project case study confirmed that the system's CPM calculation of 73 days aligns with manual and Microsoft Project results, demonstrating the model's accuracy and reliability in determining the minimum project duration. However, differences observed in non-critical activities highlight the need for integrating more complex scheduling considerations, such as calendar adjustments, resource leveling, and buffer time. Future research is suggested to enhance this system by incorporating automatic report export features in PDF or Excel formats, developing multi-user and database-integrated environments for collaborative project management, and optimizing the mobile interface for better usability in field operations. Furthermore, integrating artificial intelligence for predictive analysis of project delays could make this tool a more comprehensive and intelligent decision-support system for construction project planning and control.

REFERENCES

Alaloul, W. S., Liew, M. S., & Zawawi, N. A. W. A. (2016). Identification of coordination factors affecting building projects performance. *Alexandria Engineering Journal*, *55*(3), 2689–2698.

Bishnoi, N. (2018). Critical Path Method (CPM): A coordinating tool. *International Journal of Engineering Research*, *9*(1), 12–18.

Dwiretnani, A., & Kurnia, A. (2018). Optimizing project implementation with the CPM (Critical Path Method) method. *Journal of Civil Engineering*, *1*(2), 58. https://doi.org/10.33087/talentasipil.v1i2.8

Ervianto, W. I. (2005). *Construction project management* (Latest Edition). Yogyakarta: Andi Publishers.

Iwawo, E. R. M., Tjakra, J., & Pratasis, P. A. K. (2016). Application of the CPM method in construction projects (Case study of the construction of a new building of the Eben Haezar Manado complex). *Journal of Static Civility*, *4*(7), 421–430.

Kansha, W. M. (2023). Comparative analysis of the structure and performance of the CodeIgniter and Laravel frameworks in web application development. *Journal of Information Technology*, *9*(1), 15–25.

Kerzner, H. (2017). *Project management: A systems approach to planning, scheduling, and controlling* (Twelfth edition). Wiley.

Morris, P. W. G. (2021). *The management of projects*. Thomas Telford Publishing.

Pamungkas, E. R., Susanti, D., & Resmanah, D. (2020). Web-based waste bank application in Teja village. *Unisbank Journal of Informatics*, *8*(2), 79–88.

Pattiraja, A. H., Gai, A. M., Pontan, D., & Sahrullah. (2024). *Civil engineering project management: Strategies and tactics for construction projects* (Print I). PT Media Penerbit Indonesia.

Polii, R. B., Walangitan, D. R. O., & Tjakra, J. (2017). Time control system with Critical Path Method (CPM) in construction projects. *Journal of Static Civility*, *5*(4), 245–252.

Ramadhan, W., & Putra, S. H. (2022). The web-based attendance application for students and lecturers at Ganesha Medan Polytechnic uses PHP and MySQL. *Remik*, *6*(3), 526–533. https://doi.org/10.33395/remik.v6i3.11674

Rawat, A. (2020). A review on Python programming. *International Journal of Research in Engineering and Science*, *4*(9), 45–52.

Saabith, S., Vinothraj, T., & Fareez, M. (2021). A review on Python libraries and IDEs for data science. *International Journal of Research in Engineering and Science*, *8*(2), 23–31.

Tan, J., Chen, Y., & Jiao, S. (2023). Visual Studio Code in introductory computer science course: An experience report. *arXiv preprint* arXiv:2303.10174. https://doi.org/10.48550/arXiv.2303.10174

Wasito, W., & Syaikhudin, A. Y. (2020). Study on the application of the Critical Path Method (CPM) in the construction project of PT Semen Gresik Rembang cement factory. *J-MACC: Journal of Management and Accounting*, *3*(2), 74–91. https://doi.org/10.52166/j-macc.v3i2.2072