
Design And Development Of A Web-Based Event Organizer Management Application At CV. Cipta Abadi Organizer Using The Extreme Programming Method

Dwi Rama Rizki Santoso¹⁾, Yudhi Raymond Ramadhan²⁾, Muhammad Rafi Muttaqin³⁾
^{1,2,3)}Wastukancana College of Technology

*Corresponding Author

E-mail: dwirama38@wastukancana.ac.id

Abstract

The increasing complexity of event organizer operations requires an integrated management application to improve administrative efficiency, coordination, and activity monitoring. This study aimed to design and develop a web-based Event Organizer Management application for CV. Cipta Abadi Organizer using the Extreme Programming (XP) method. This research employed a Research and Development approach with the Extreme Programming software development methodology. The population comprised all event management activities at CV. Cipta Abadi Organizer, while the sample focused on operational processes, including client management, vendor management, event scheduling, and event progress monitoring. Data were collected through observation and interviews, while data analysis was conducted by identifying user requirements, modeling the system using Unified Modeling Language (UML), implementing the application with Laravel and MySQL, and evaluating functionality through Test-Driven Development testing. The results indicate that the developed application successfully integrates client, vendor, event, rundown, and field log management, while all tested features function according to user requirements. It also improves data organization, operational coordination, and monitoring efficiency. In conclusion, the developed application is suitable for supporting integrated event organizer operational management and provides a practical digital solution for improving organizational performance.

Keywords: Event Management System, Extreme Programming, Management Application, Laravel, Web Application

INTRODUCTION

Advances in information technology have encouraged various business sectors to utilize web-based applications to improve operational efficiency and service quality. In the event organizer industry, the use of event management applications is essential because they can integrate client, schedule, team, and vendor data management, making information more structured, easily accessible, and minimizing recording errors (Fikria et al., 2025).

The complexity of event management demands a system capable of supporting integrated planning, coordination, monitoring, and evaluation. Event management applications help accelerate scheduling, reporting, and data analysis, thereby improving work efficiency and service quality (Rahmatullah & Sanjaya, 2025).

Based on interviews at CV. Cipta Abadi Organizer, client data and event information are still managed using Microsoft Word and manual recording. This creates difficulties in data retrieval, risks data loss, and limits the ability to monitor event preparation progress, making the administrative process less effective.

To address these issues, a web-based Event Organizer Management application is needed that can integrate data management and activity monitoring. This research applies the Extreme Programming (XP) method because it can produce flexible software through an iterative and collaborative development process (Fachrurrozi et al., 2025).

Previous research has shown that web-based event organizer information systems can improve the efficiency of data management and activity monitoring (Rizky et al., 2021; Marpaung et al., 2025). However, research focusing on internal operational management of event organizers is still limited. Therefore, this study aims to design and build a web-based event organizer management application for CV. Cipta Abadi Organizer using the Extreme Programming method. The novelty of this research lies in the development of an application that integrates client and vendor management,

preparation stages, and activity progress monitoring into a single system that supports comprehensive internal operations.

RESEARCH METHODS

This study uses the Extreme Programming (XP) method as a software development method to design and build a web-based Event Organizer Management application for CV. Cipta Abadi Organizer. This method was chosen because it supports an iterative, collaborative, and flexible development process according to user needs. The stages involved include planning, design, coding, testing, and release (Fachruruzi et al., 2025).

The data collection instruments used in this study consisted of observation and interviews. Observations were conducted by directly observing the ongoing Event Organizer management process at CV. Cipta Abadi Organizer to identify workflows and problems encountered. Interviews were conducted with parties involved in event management to obtain the system requirements to be developed. The data obtained were then analyzed as a basis for compiling the application's functional requirements, which were then translated into a system design using the Unified Modeling Language (UML) which includes Use Case Diagrams, Activity Diagrams, Sequence Diagrams, and Class Diagrams. The application was developed using the PHP programming language with the Laravel framework and MySQL database, then tested to ensure that each system function runs according to user needs.

The population in this study was the entire activity management process at CV. Cipta Abadi Organizer, while the research sample focused on operational management processes, including client data management, activity scheduling, vendor coordination, and monitoring event preparation progress. The sample selection was adjusted to the scope of the study, which focused on the development of a web-based Event Organizer management application.

The research procedure begins with observation and interviews to obtain information about ongoing business processes and user needs. Next, a planning phase is conducted to identify software requirements and determine the scope of the system to be developed. The next phase is design, which involves designing the system using UML and developing the application interface. After the design process is complete, coding is carried out using PHP, the Laravel framework, and MySQL as the database. The developed application then enters the testing phase to ensure all functions run as required and to correct any errors. The final phase is release, which involves implementing the application after it has been declared fit for use. The entire research process is then documented in a research report (Fachruruzi et al., 2025).

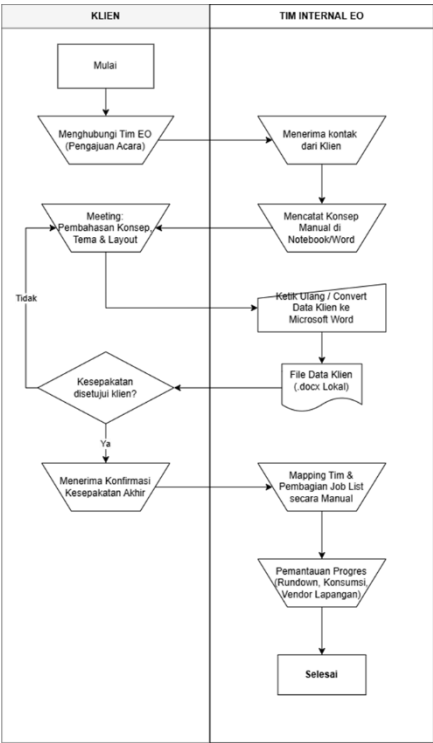
RESULTS AND DISCUSSION

Observation & Interview

In the initial stages of Extreme Programming, the author conducted observations and interviews at CV. Cipta Abadi Organizer to identify system issues and requirements. The results showed that data management was still manual, complicating data retrieval and activity monitoring, and increasing the risk of data loss.

Based on the interview results, a web-based application was needed that could manage client data, schedules, vendors, and event progress. These needs were then formulated into User Stories as the basis for application development.

FlowmapRunning System



Picture 1. Flowmap of Running System

Use Case Diagram

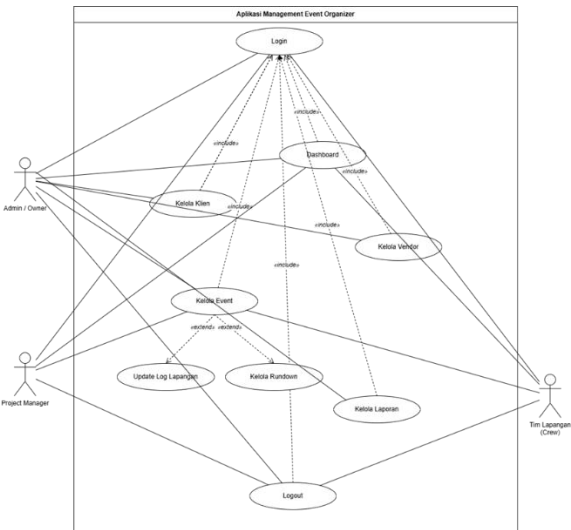


Figure 2. Use Case Diagram

Scenario Diagram

The use case scenario in the Event Organizer Management application includes eight main functions. All actors, namely Admin, Project Manager, and Crew, can log in to access the system according to their access rights and log out to end the application usage session. After successfully logging in, users can access the dashboard that displays a summary of operational data and the latest event information. Admin has the right to manage client and vendor data, including the process of adding, changing, and deleting data. Furthermore, Admin together with the Project Manager can manage event data, including information on clients, vendors, responsible parties, and event implementation status, and manage rundowns as activity implementation schedules. In addition, Admin, Project Manager, and Crew can update field logs to convey activity developments during the event implementation. At each process, the system validates data before saving changes, while if the data is invalid or the user cancels the process, the system will display an error message and will not

save the changes.

Class Diagram

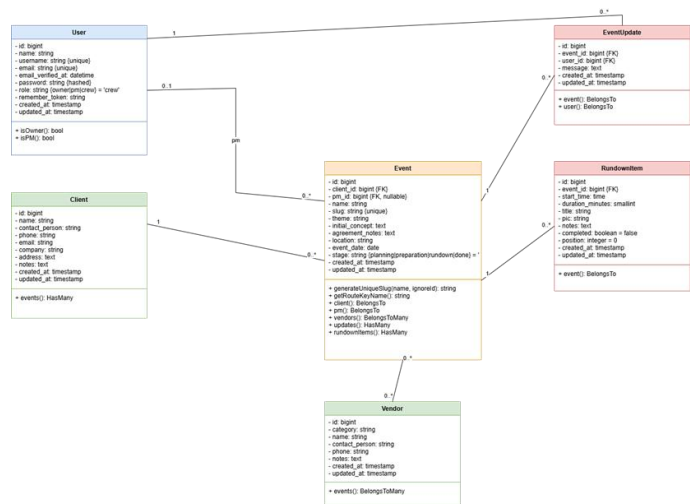


Figure 3. Class Diagram

Class Diagram The Event Organizer Management application consists of six main classes, namely User, Client, Event, Vendor, EventUpdate, and RundownItem. The User class manages user data along with access rights as Owner, Project Manager, or Crew. The Client class stores client data that has a one-to-many relationship with Event. The Event class is the center for managing event data connected to Client, User as Project Manager, Vendor, EventUpdate, and RundownItem. The Vendor class manages service provider data that has a many-to-many relationship with Event. Furthermore, the EventUpdate class is used to record activity progress or field logs based on events and the users who create them, while the RundownItem class functions to manage a structured event activity schedule that is connected to each event. All of these classes are interrelated to support data management, coordination, and monitoring of Event Organizer activities in an integrated manner.

Wireframe

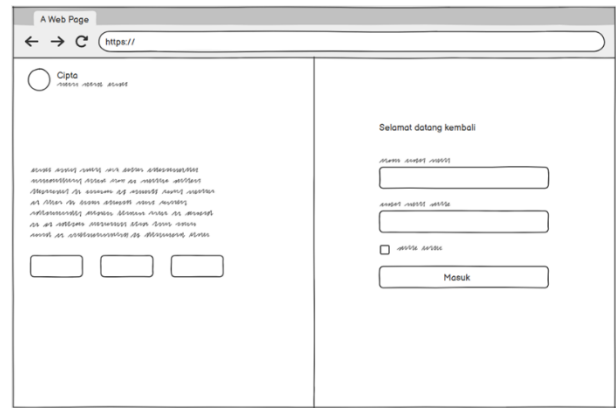


Figure 4. Wireframe Login

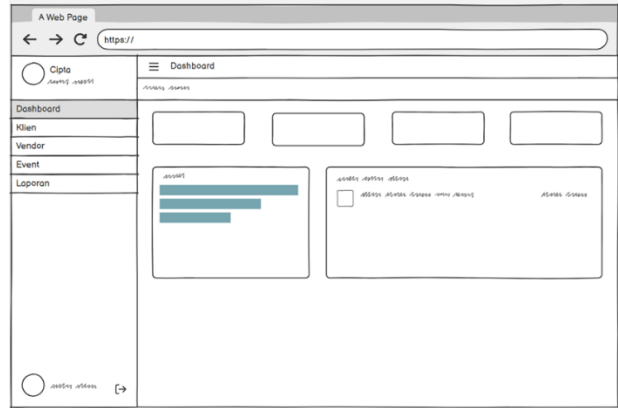


Figure 5. Wireframe Dashboard

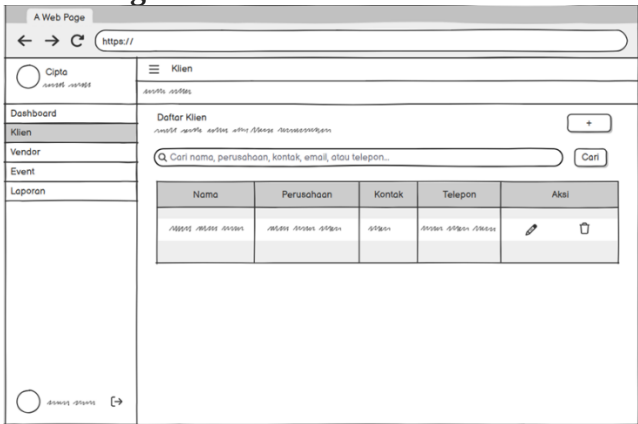


Figure 6. WireframeClient List

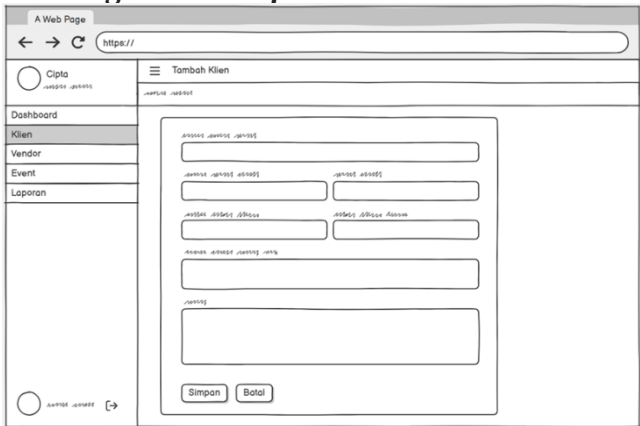


Figure 7. WireframeAdd Client

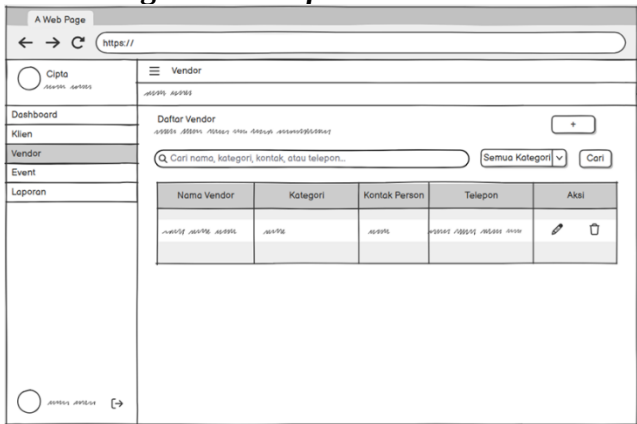


Figure 8. WireframeVendor List

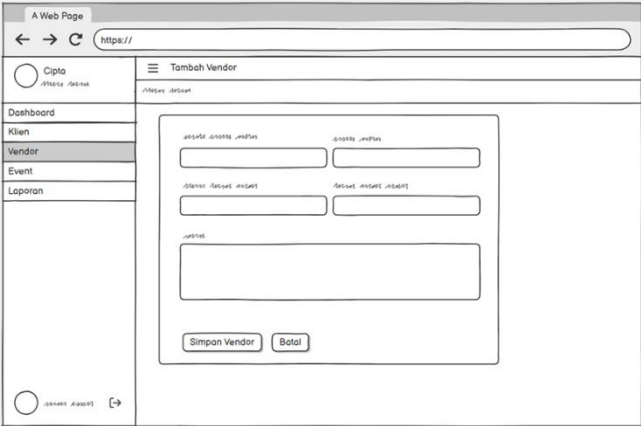


Figure 9. WireframeAdd Vendor

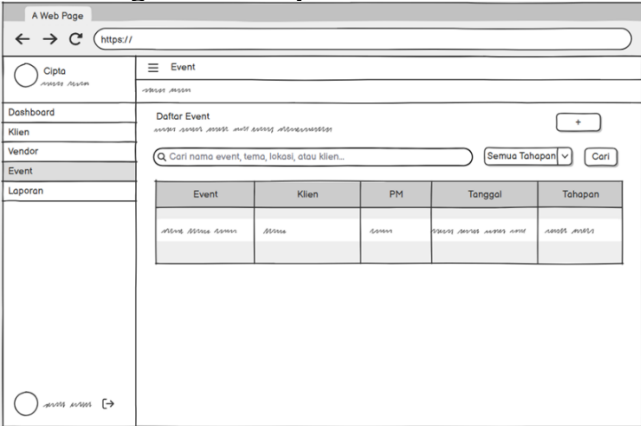


Figure 10. WireframeListEvent

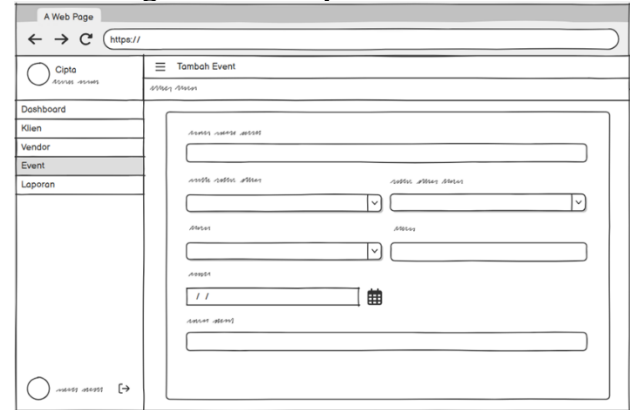


Figure 11. WireframePlusEvent

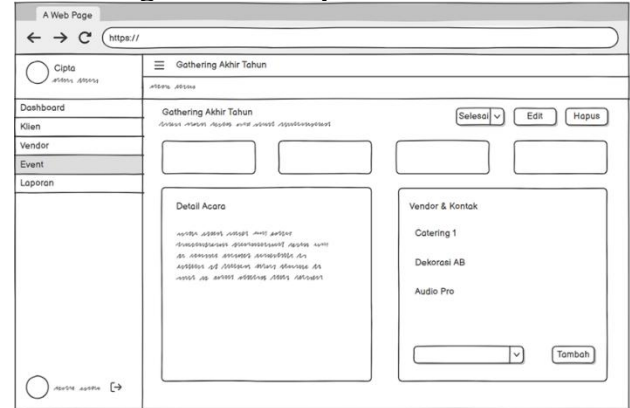


Figure 12. Wireframe Detail Event

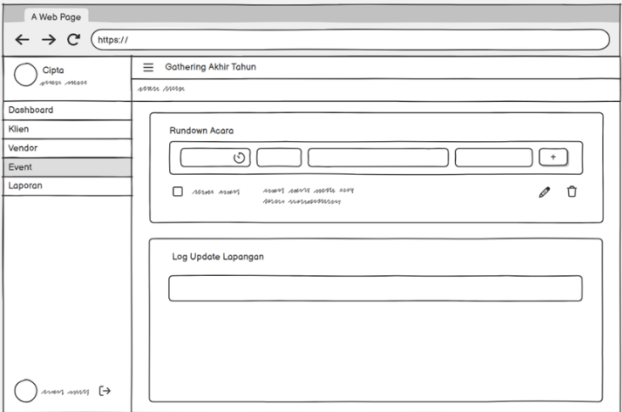


Figure 13. Wireframe Rundown & Update Log



Figure 14. Report Menu Wireframe

Coding

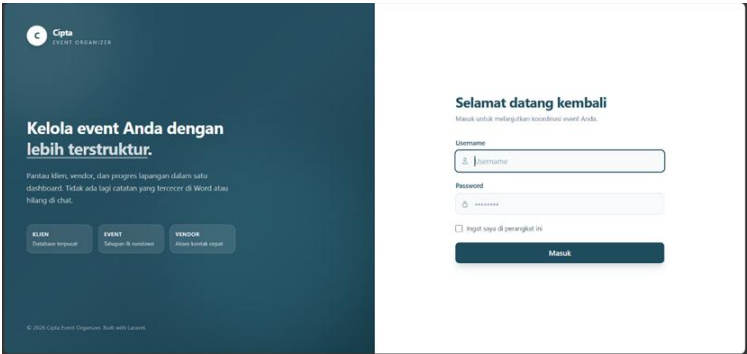


Figure 15. InterfaceLogin

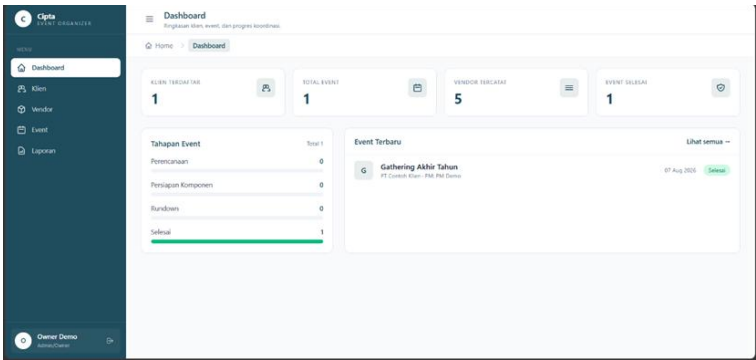


Figure 16. InterfaceDashboard

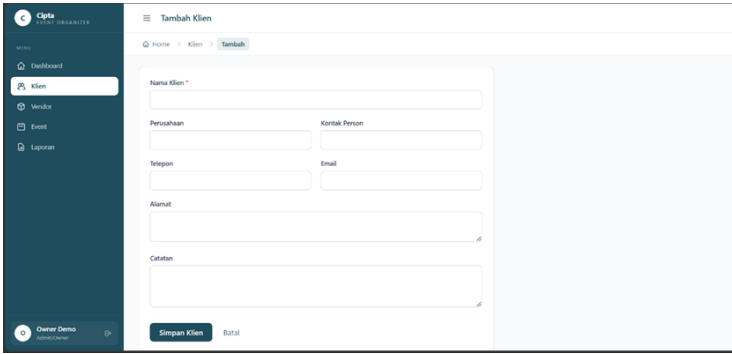


Figure 17. Add Client Interface

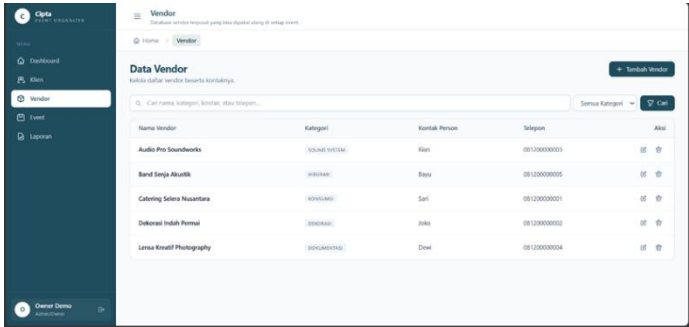


Figure 18. Vendor List Interface

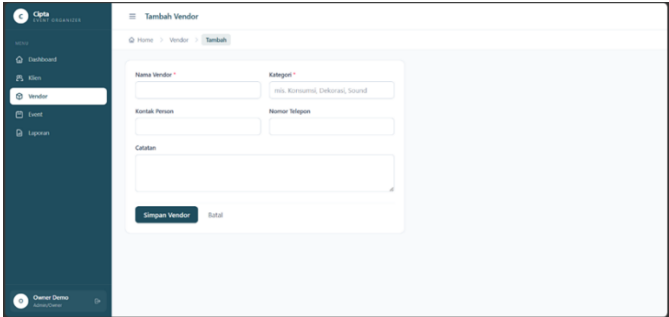


Figure 19. Add Vendor Interface

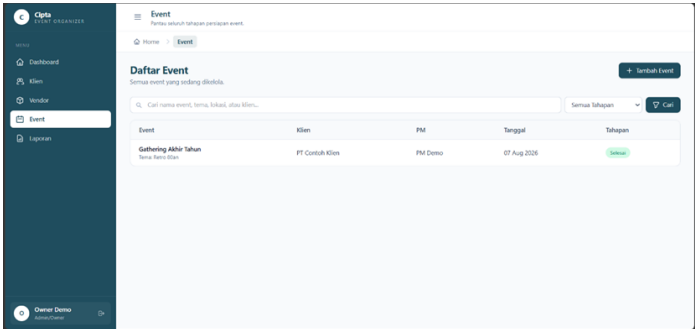


Figure 20. List InterfaceEvent

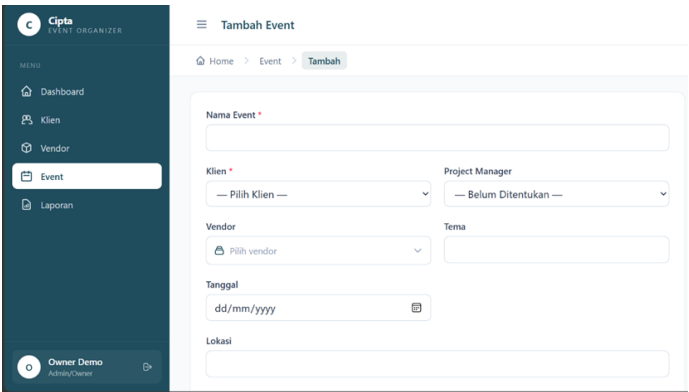


Figure 21. Add InterfaceEvent

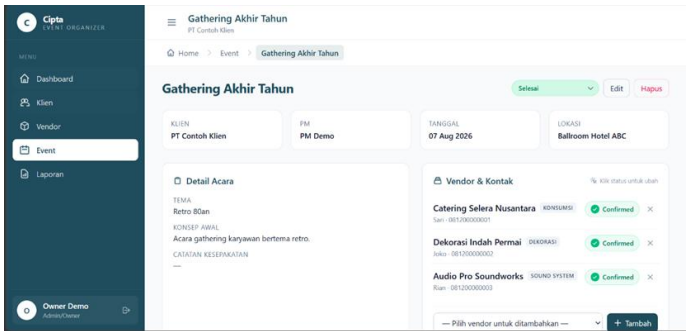


Figure 22. Detail InterfaceEvent

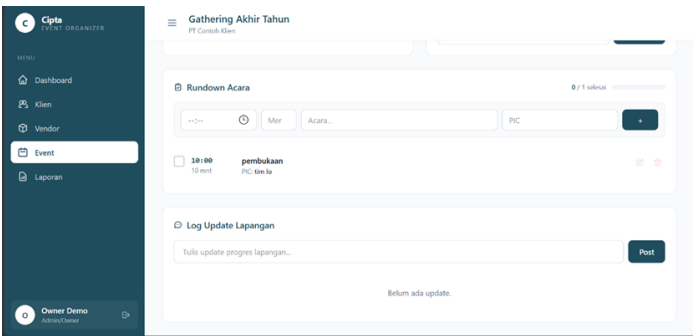


Figure 23. InterfaceRundown & Log Update

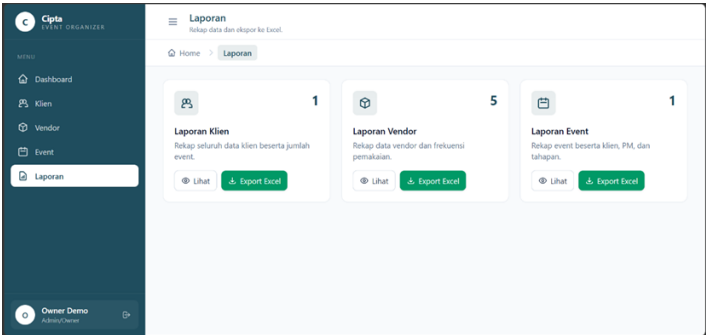


Figure 24. Report Menu Interface

Testing

```
PASS Tests\Feature\RoleAccessTest
✓ it melarang crew membuka form tambah klien 0.44s
✓ it melarang crew membuat event 0.02s
✓ it melarang PM mengakses laporan 0.02s
✓ it mengizinkan owner mengakses laporan 0.03s
✓ it melarang PM membuka event milik PM lain 0.03s
✓ it mengizinkan PM membuka event miliknya sendiri 0.03s
✓ it mengizinkan owner membuka event milik PM mana pun 0.03s

Tests: 7 passed (7 assertions)
Duration: 0.82s
```

Figure 25. Greentick Test-Driven Development Cycle

```
FAIL Tests\Feature\RoleAccessTest
✗ it melarang crew membuka form tambah klien 1.48s
✗ it melarang crew membuat event 0.50s
✗ it melarang PM mengakses laporan 0.02s
✓ it mengizinkan owner mengakses laporan 0.02s
✗ it melarang PM membuka event milik PM lain 0.03s
✓ it mengizinkan PM membuka event miliknya sendiri 0.02s
✓ it mengizinkan owner membuka event milik PM mana pun 0.09s

FAILED Tests\Feature\RoleAccessTest > it melarang crew membuka form tambah klien
Expected response status code [403] but received 200.
Failed asserting that 200 is identical to 403.
```

Figure 26. Red Test-Driven Development Cycle

Table 1 TestingTest-Driven Development

No	Test Case	Red Cycle (Fail)	Green Cycle (Succeed)	Refactor Cycle
	Users can login and be directed to the Dashboard	Route `dashboard` does not exist yet	After creating `AuthController` and authentication route	Directs to Dashboard according to access rights and adds session validation
	Login failed with incorrect credentials	No error message is displayed	After adding validation to `AuthController`	Error messages are written in Indonesian
	Logout ends the session and returns to the login page.	Session not deleted after logout	After implementation of `logout()` (invalidate session + token)	Added CSRF token regeneration after logout
	Crew are prohibited from opening the add Client form	Crew can still open the form (response 200)	After installing the `role:owner` middleware on the client route	Creating a reusable `EnsureUserHasRole` middleware
	Crew is prohibited from creating events	Crew can still save events	After restricting the `events.store` route with `role:owner`	Grouping route owners in one middleware group
	PM is prohibited from accessing the Report menu	PM can still open report (response 200)	After moving the `reports` group to the `role:owner` middleware	Hiding the Reports menu from the PM sidebar via the `@role` directive

No	Test Case	Red Cycle (Fail)	Green Cycle (Succeed)	Refactor Cycle
	Admin can add Client data	Error `Column 'company' not found` (schema not yet appropriate)	After adjusting migration & `fillable` Client model	Added `required` validation to client name
	Admin can change and delete Clients	The `update`/`destroy` method is not yet available	After completing the `ClientController` (resource)	Give status notification after save/delete
	Admin can add Vendors	Vendor category validation is not working	After adding validation rules in `VendorController`	Sort vendors by category then name
0	PM only sees his own Events	PM can open events owned by other PMs (response 200)	After adding scope `Event::visibleTo()` & `isVisibleTo()`	Apply the same scope to Dashboard, Rundown, Vendor, and Log
1	Rundown can be added in the correct order	The `position` column is not auto-populated	After calculating the `position` of the last item + 1	Sort rundown by `position` then `start_time`
2	Field Logs are saved with author & time	Empty messages remain saved	After validation of `message` `required` in `EventUpdateController`	Automatically logs `user_id` and displays the latest timeline
3	Reports can be exported to Excel files	Excel file failed to form (incorrect column format)	After implementing export with PhpSpreadsheet (`*.xlsx`)	Add title, agency header, and column format (date/number)

Release

In the Release phase, the web-based Event Organizer Management application that has been developed and passed testing is implemented at CV. Cipta Abadi Organizer. The application is deployed to a hosting server using a MySQL database so that it can be accessed via the internet. This phase includes server configuration, initial data migration such as client and vendor data, and user account creation according to Admin, Project Manager, and Crew access rights. Once all functions are confirmed to be running properly, the application is ready to be used to support data management and event organization operations.

CONCLUSION

Based on the research results, it can be concluded that the web-based Event Organizer Management application developed using the Extreme Programming (XP) method successfully met the operational needs of CV. Cipta Abadi Organizer. The application is able to integrate the management of client, vendor, event, rundown, and field log data into one system so that the administration, coordination, and monitoring processes of event implementation become more structured and efficient compared to the previous manual process. The test results show that all main features, such as user authentication, data management, access rights restrictions, rundown management, field log recording, and report generation, have run according to user needs. Implementation of the application in a production environment also shows that the system is ready to be used to support the company's operational activities more effectively.

This research still has limitations because the application development and implementation were only carried out at one company, namely CV. Cipta Abadi Organizer, so the results cannot be generalized to the characteristics of different Event Organizers. In addition, the application still focuses on internal operational management and does not yet support other features, such as automatic notifications, integration with third-party services, or more comprehensive data analysis. Therefore, further research is recommended to develop more complete features, expand the implementation to different research objects, and evaluate user satisfaction and system performance after use for a certain period. This development is expected to increase the benefits of the application and serve as a reference for the development of a more adaptive and integrated Event Organizer management information system.

REFERENCES

- Fachrurozi, A., Fitriana, LA, Faddillah, U., & Sugiyarto, I. (2025). Implementation of Extreme Programming in the creation of the PT Naga Emas Internasional E-Accountant information system website. *REMIK: Research and E-Journal of Computer Informatics Management*, 9(1), 73–87. <https://doi.org/10.33395/remik.v9i1.14294>
- Fikria, N., Dinanti, RP, & Siahaan, ATAA (2025). Designing an event management website as a solution for managing activities in the Public Welfare Section of the Deli Serdang Regent's Office. *Scientific Journal of Mechanical, Electrical and Computer Engineering*, 5(3), 302–311. <https://doi.org/10.51903/juritek.v5i3.5976>
- Marpaung, MNA, et al. (2025). Design of a web-based campus event processing information system using the Waterfall method. *National Journal of Computer Technology*, 5(3), 482–494.
- Rahmatullah, A., & Sanjaya, FI (2025). Analysis of event management application development in Event Organizers using the Rapid Application Development model. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 5(1), 216–221. <https://doi.org/10.57152/malcom.v5i1.1729>
- Rizky, MF, Cleopatra, M., & Parwatiningtyas, D. (2021). Java-based event organizer information system at Wahana Entertainment. *Journal of Information System, Applied, Management, Accounting and Research (JISAMAR)*, 5(1), 117–124. <http://journal.stmikjayakarta.ac.id/index.php/jisamar>