

Development of a Sustainable Information System Using Structured Routing and Livewire Methods in Educational Institutions in Jakarta

Adi Fitrianto

Information System Program, Faculty Of Computer, Universitas Media Nusantara Citra, Indonesia
adi.fitrianto@mncu.ac.id



*Corresponding Author

Article History:

Submitted: 22-07-2024

Accepted: 23-07-2024

Published: 31-07-2024

Keywords:

Livewire Methods; Structured Routing; Educational Institutions.

Brilliance: Research of Artificial Intelligence is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

ABSTRACT

This research aims to develop a data management application for instructors and training programs at MNC Academy. The underlying issue for developing this application is the lack of an efficient and practical data recording system and insufficient accessibility to important information about instructors and training. Instructor and training data are often complex and continually evolving, making manual management or using office software inefficient and time-consuming. Additionally, digital data recording of instructors and training is needed for the promotional website of MNC Academy, ensuring organized and timely access to relevant information for the public. The application is developed using PHP programming language with the Laravel framework as its base. The application also employs structured routing and Livewire technology to build mini components that support the system's functionality more effectively. Structured routing is an approach to clearly and systematically organize routes, enabling more systematic management of URL links and providing a structured way to handle navigation routes. Livewire in the Laravel framework is a technology that allows the creation of interactive user interface components without the need to write separate JavaScript code. Using Livewire helps simplify the development of dynamic components by reducing technical complexity. With the implementation of this data management application, it is expected that the data of instructors and training programs at MNC Academy can be managed more systematically, enabling neat record-keeping and providing easier access for those needing this information.

INTRODUCTION

MNC Academy, a training institution owned by MNC Group, has become a foundation for innovation and quality education. One critical aspect of its operations is the management of data for instructors and training programs. This data is used not only for internal operational purposes but also as a key part of the institution's promotional and development strategy. Instructor-related data is not limited to basic information but also includes complex details such as portfolios, work experience, teaching history, and more. Similarly, training data encompasses a variety of information that requires meticulous recording.

Currently, data management for instructors and training programs relies on MS Word applications. This usage is ineffective given the continuously increasing and complex nature of the data. Additionally, inputting this data into the promotional website would be time-consuming if done individually based on the information stored in MS Word applications.

Problem Statement

Based on the above issues, the problem statements are as follows:

- How to design a data management system for instructors and training programs.
- How to build a web-based data management system for instructors and training programs.

Research Objectives

Based on the above problems, the objectives of this research are:

- To design a data management system for instructors and training programs.
- To build a web-based data management system for instructors and training programs.



Research Benefits

By designing and implementing an integrated web-based data management system for instructors and training programs, this research aims to address the challenges faced by the operational team regarding complex data management processes. The designed system is expected to provide significant benefits such as increased operational efficiency by reducing reliance on manual records, better and real-time data accessibility, reduction of errors and data duplication through comprehensive integration, and improved service quality with structured data.

Research Scope

This research focuses on designing and developing a web-based management system for instructor and training data at MNC Academy. The research includes analysis and development processes.

The structured routing method and Livewire technology are used in the creation of this data management system. Structured routing is adopted because it can systematically organize data distribution paths. This allows for efficient data management, emphasizing organized data flow arrangement, helping to optimize performance and information processing in the data management system (Madani, Munti, & Gusman, 2023). Additionally, Livewire technology is used in building this system because it can create reusable components and real-time speed (Daru & Adhiwibowo, 2021), thereby saving time and effort in web application development.

The use of structured routing and Livewire methods to develop the data management system for instructors and training at MNC Academy demonstrates the adoption of modern technology in education and information management. The proposed information system not only involves digital data processing but also includes the use of the latest methods and technologies in managing information related to instructors and training. This reflects the transition from manual processing to a digital-based approach involving software and modern technologies in data management, related to the digital economy aspect. By leveraging technology to enhance efficiency and effectiveness in data management, this research highlights the adoption of technological innovations that form the foundation of the growing digital economy.

The use of Livewire in the development of this system has significant implications for sustainability. Livewire enables the creation of reusable digital components, effectively reducing resource waste in the software development process. By promoting modularity and reusability, Livewire reduces code duplication and simplifies application maintenance. This approach not only enhances development efficiency by reducing the need to write and maintain repetitive code but also reduces the energy and time consumption associated with software creation. In this way, Livewire supports more sustainable and environmentally friendly software development practices, contributing to more efficient resource management and reducing the environmental impact of the development process.

LITERATURE REVIEW

The management of instructor and training data is carried out through the development of an application using the PHP programming language under the Laravel framework. PHP is a programming language that acts as a server-side script to develop websites by embedding code into HTML documents (Madani, Munti, & Gusman, 2023). The use of PHP in application development is recommended together with the Laravel framework because this framework offers several excellent features, including a template engine, routing management, and modularity, which significantly support developers in building applications (Daru & Adhiwibowo, 2021).

Structured routing in Laravel refers to the systematic and organized definition of web application routes. This allows developers to assign specific URLs to appropriate actions or logic within the application. In this context, Laravel provides a clear and structured way to determine how HTTP requests are handled, allowing the grouping of control logic based on the accessed URL (Novianto & Maryam, 2022). Through this routing system, developers can easily manage the navigation of user requests to various parts of the web application built using Laravel (Ambriani & Nurhidayat, 2020).

In the Laravel framework, Livewire is a technology that enables dynamic PHP-based user interface development without requiring deep knowledge of JavaScript. It allows the use of interactive components in web application creation by using PHP to directly interact with frontend elements without manually writing JavaScript code (Supriyanto & Susanto, 2019). Livewire facilitates the creation of responsive and dynamic applications by integrating PHP backend logic with user interface elements, enabling Laravel developers to create engaging web applications more quickly and easily.

The use of Livewire technology is prevalent in the development of web-based systems such as news management tracking systems (Yudhistira, Setyawan, & Harsiti, 2022), inventory management information systems (Daru & Adhiwibowo, 2021), and company profile application systems (Firmansyah & Zen, 2023), among others. Livewire can facilitate the creation of simple, dynamic interfaces and reusable components. Structured routing in the Laravel framework offers significant benefits in web application development. One such benefit is its ability to clearly manage application navigation (Pasaribu, Nurhayati, & Windasari, 2022). By leveraging this structured routing mechanism, developers can easily determine how URLs will be directed to various actions (controllers) within the application. This helps create a structured and cohesive layout for web application users. Routing also allows for the separation of

application control logic from route definitions (Zy, 2022), facilitating efficient management and maintenance of application code. With these benefits, structured routing in Laravel not only simplifies user navigation but also provides a robust framework for managing the complexity of web applications more efficiently and systematically.

Developing systems with the theme of managing instructor and training data using structured routing and Livewire is still limited. Therefore, based on the understanding previously outlined, this system will implement structured routing and Livewire as the main foundation. The reason for choosing these methods lies in their ability to clearly and systematically manage navigation within the application and create reusable components. By utilizing structured routing and Livewire, this application is expected to provide a responsive user experience, enhance the efficiency of instructor and training data management at MNC Academy, and serve as a solid foundation for the growth of this educational institution.

Further research has shown that using web-based systems for data management in educational institutions leads to significant improvements in efficiency and effectiveness. For example, the development of web-based student data management systems has demonstrated substantial benefits in managing student information and facilitating administrative processes (Widya Ayu Suryaningtyas, Adam Hendra Brata, & Lutfi Fanani, 2020). Similarly, the design of web-based management systems for schools and educational institutions has shown a positive impact on the management and accessibility of data (Hakiki et al., 2021). These findings support the notion that implementing structured routing and Livewire can significantly improve the management of instructor and training data.

Moreover, the integration of web-based training management systems in educational institutions has been found to streamline training processes and improve overall operational efficiency (Agus, Shelve Nidya Neyman, & Hendra Rahmawan, 2023). The development of such systems provides a robust platform for managing training programs, tracking progress, and ensuring the availability of up-to-date information for both instructors and trainees (Rofiq, Sholiq, & Muqtadiroh, 2013). This underscores the potential benefits of employing structured routing and Livewire in developing a sustainable information system for educational institutions in Jakarta. Additionally, the development of web-based management systems for managing teacher and student data has proven to enhance the efficiency and effectiveness of data management in educational institutions (Sofyan, Sadikin, Sefriyandi, & Djutalov, 2023).

METHOD

This research employs the Systematic Literature Review (SLR) method to validate theories and determine solutions. The SLR method is used with the aim of reviewing the case study thoroughly before proceeding with further processes. Additionally, SLR is conducted to gather knowledge from previous studies regarding benefits, proposed concepts, implementation challenges, technology utilization, and updates from similar research. The steps or workflow of the systematic literature review are illustrated in Figure 1.

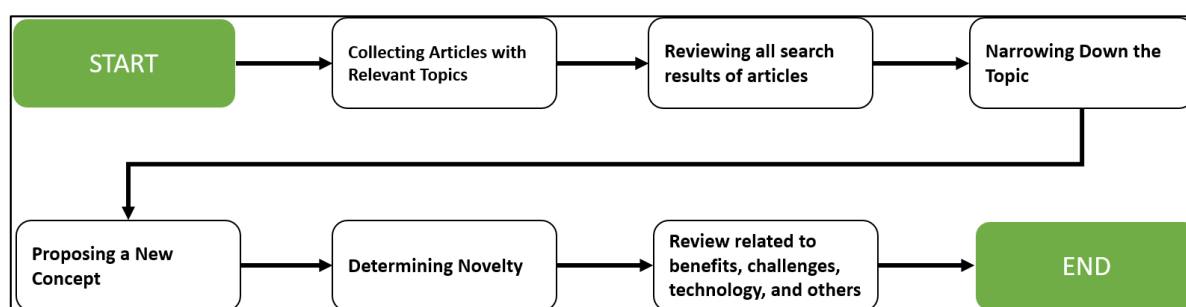


Figure 1. SLR Workflow

Stages of Systematic Literature Review

The first step involves collecting articles using keywords related to data management systems. Each collected article is then reviewed. Following this, further articles related to the topics of Structured Routing and Livewire are gathered by narrowing the focus. A subsequent review is performed to uncover previously unexplored aspects to develop similar studies.

Research Stages

This research comprises several stages, beginning with data collection related to operational conditions, followed by a literature review, application design, coding, and final testing of the application. The stages of the research are outlined in Figure 2 below.

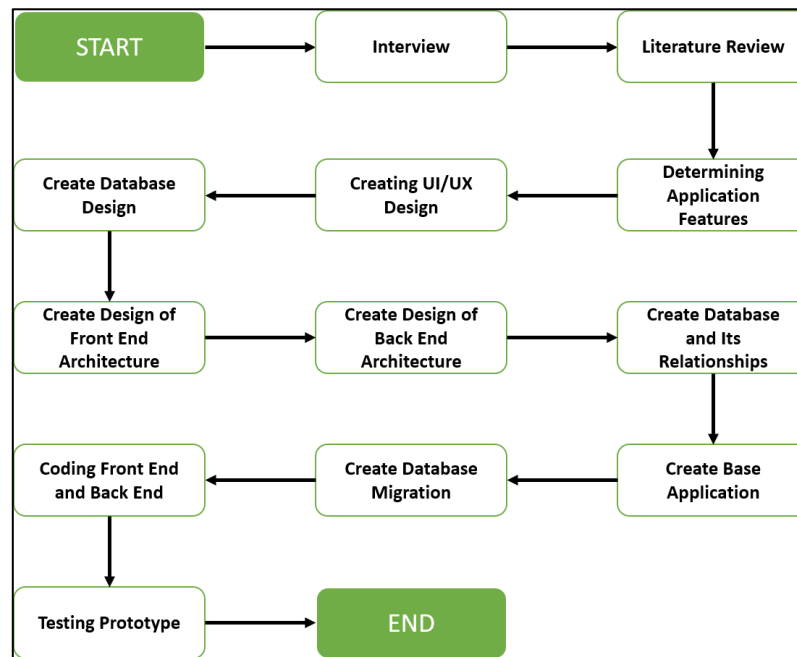


Figure 2. Research Stages

Based on the diagram, the research includes several stages. Here is a description of each stage in this research:

1. Interview

This stage is the initial step of the research involving interviews with several operational staff at MNC Academy. The purpose of these interviews is to gather information about the challenges related to managing and recording instructor and training data, which will be transitioned to a digital format.

2. Literature Review

Following the interviews, a literature review is conducted to evaluate the feasibility of using Structured Routing and Livewire methods in developing the data management application. This literature review aims to identify aspects that have not been previously discussed to advance similar studies.

3. Determining Application Features

In this stage, the author creates a list of features based on the needs identified during the interviews regarding the management of instructor and training data. The features are also based on the capabilities of the framework applied to the application.

4. Designing UI/UX

After the literature review, the UI/UX design or the application's visual layout is created. This process uses tools such as Figma and Canva. UI design is completed before coding since the front-end coding will refer to the interface design.

5. Designing the Database

This stage involves designing the database, including tables as entities and their attributes according to the application's needs. Besides the tables and attributes, the relational design between tables is also created to assist programmers in implementing the database.

6. Designing Front-End Architecture

At this stage, the architecture for the front-end is designed. Unlike UI design, front-end architecture involves determining the layout of PHP Blade files, CSS, JavaScript, and folder structures within the Laravel framework.

7. Designing Back-End Architecture

This stage involves designing the back-end architecture, which includes elements such as controllers, Livewire, routes, models, and migrations. Each part plays a distinct role. Additionally, folder structures and file placements are designed to facilitate the development process.

8. Creating the Database and Implementing Relations

In this stage, the database is created using the MySQL Database Management System (DBMS). The database and its relationships are implemented according to the database design from Stage 5.

9. Developing the Base Application

Once the database and its relations are set up, the base application is developed using the Laravel framework and PHP Artisan as the command-line interface. This step accommodates the overall coding process.

10. Integrating Base Application and MySQL Database

When the DBMS and base application are ready, integration between the MySQL database and the base application is performed using database migration and MySQL connector within the Laravel environment.

11. Front-End and Back-End Coding

After the database and base application integration, the next stage involves coding the application. This includes creating controllers, models, routes, Livewire components, import, and export functionalities using PHP. Front-end coding involves PHP Blade templates, CSS, and JavaScript.

12. Application Testing

Testing is performed after the application and database are fully developed. The testing includes black box, white box, and dynamic testing. If bugs are found during this stage, developers will perform bug fixing for any errors identified.

Raw User Interface Design

The raw user interface design for the instructor and training data management application to be developed in this research is as follows. This design is a preliminary or draft version, providing an overview of the application that will be produced from this research proposal.

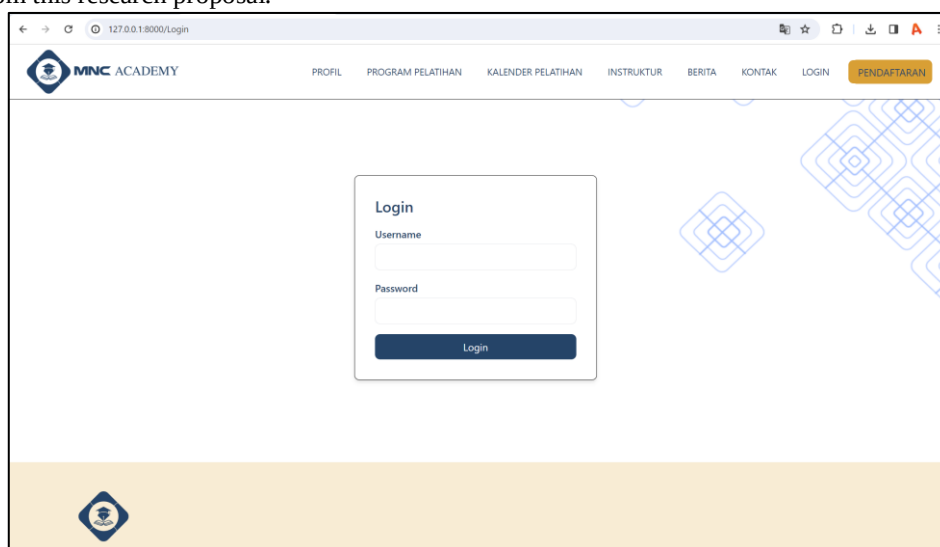


Figure 3. Raw User Interface Design

Figure 3 illustrates the login page design for the MNC Academy data management system. The plan includes separate mini components for the navbar and login card containing input fields for username and password, created using Livewire.

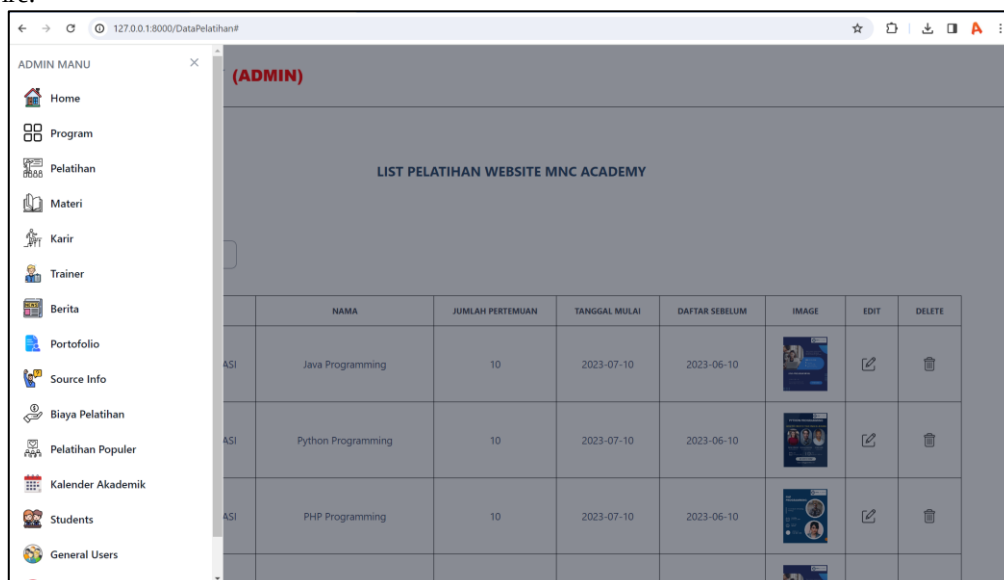


Figure 4. Raw Interface for Training Data Management Menu

The application will feature several components, including login, user data management, instructor data management, and training data management. These features will be developed according to user needs.

RESULT

This system design includes two main features: instructor data management and training data management. The technologies of structured routing and Livewire have been successfully applied to these two features.

Application of Structured Routing

The application of structured routing technology has been successfully implemented in the instructor data management feature. Through the programming code written according to the Laravel framework's guidelines, the routing operations related to the application pages designed specifically for the instructor data management feature have been neatly organized and functioned smoothly. The structured routing code is implemented in the web.php file, as shown in Figure 5 below.

```

166 //
167
168 //program role
169 Route::get('/ProgramPelatihan/{idProgram}', [ProgramPelatihanController::class, 'getProgramPelatihan'])->name('ProgramPelatihan');
170 Route::get('/DetailPelatihan/{idPelatihan}', [ProgramPelatihanController::class, 'detailPelatihan'])->name('DetailPelatihan');
171 Route::get('/DetailPelatihanByNama/{namaPelatihan}', [ProgramPelatihanController::class, 'detailPelatihanByNama'])->name('DetailPelatihanByNama');
172 Route::get('/DetailTenagaPegajar/{id}', [TrainerController::class, 'detailTenagaPegajar'])->name('DetailTenagaPegajar');
173 Route::get('/DetailPortfolio/{id}', [PortfolioController::class, 'detailPortfolio'])->name('DetailPortfolio');
174 Route::post('/insertKontakMasuk', [KontakMasukController::class, 'insertKontakMasuk'])->name('insertKontakMasuk');
175 Route::post('/insertPendaftaran', [PendaftaranController::class, 'insertPendaftaran'])->name('insertPendaftaran');
176
177 //teacher role
178 Route::view('/TeacherPage', 'TeacherPage');
179 Route::get('/TeacherUploadMateri/{namaPelatihan}', [GeneralUserController::class, 'getMateriByIdPelatihan'])->name('TeacherUploadMateri');
180 Route::post('/insertMateriByTeacher', [GeneralUserController::class, 'insertMateriByTeacher'])->name('insertMateriByTeacher');
181 Route::get('/DetailTeacher/{idTeacher}', [GeneralUserController::class, 'detailTeacher'])->name('DetailTeacher');
182
183 //student role
184 Route::view('/StudentPage', 'StudentPage');
185 Route::get('/StudentListMateri/{namaPelatihan}', [StudentController::class, 'getAllMateriByIdPelatihanForStudent'])->name('StudentListMateri');
186 Route::get('/DetailStudent/{idStudent}', [StudentController::class, 'detailStudent'])->name('DetailStudent');
187
188 //admin role
189 Route::get('/AdminIndexByRole', function () {
190     //user admin validation here
191     return view('AdminIndex');
192 });
193 Route::view('/AdminIndex', 'AdminIndex');
194
195 Route::get('/DataPelatihan', [PelatihanController::class, 'getAllPelatihan'])->name('DataPelatihan');
196 Route::post('/insertDataPelatihan', [PelatihanController::class, 'insertPelatihan'])->name('insertDataPelatihan');
197 Route::post('/updateDataPelatihan', [PelatihanController::class, 'updatePelatihan'])->name('updateDataPelatihan');
198 Route::get('/deleteDataPelatihan/{id}', [PelatihanController::class, 'deletePelatihan'])->name('deleteDataPelatihan');
199
200 Route::get('/DataMateri', [MateriController::class, 'getAllMateri'])->name('DataMateri');
201 Route::post('/insertDataMateri', [MateriController::class, 'insertMateri'])->name('insertDataMateri');
202 Route::post('/updateDataMateri', [MateriController::class, 'updateMateri'])->name('updateDataMateri');
203
204 Route::get('/DataKarir', [KarirController::class, 'getAllKarir'])->name('DataKarir');
205 Route::post('/insertDataKarir', [KarirController::class, 'insertKarir'])->name('insertDataKarir');
206 Route::post('/updateDataKarir', [KarirController::class, 'updateKarir'])->name('updateDataKarir');
207
208 Route::get('/DataTrainer', [TrainerController::class, 'getAllTrainers'])->name('DataTrainer');
209 Route::post('/insertDataTrainer', [TrainerController::class, 'insertTrainer'])->name('insertDataTrainer');
210 Route::post('/updateDataTrainer', [TrainerController::class, 'updateTrainer'])->name('updateDataTrainer');
211
212 Route::get('/DataBerita', [BeritaController::class, 'getAllBerita'])->name('DataBerita');
213 Route::post('/insertDataBerita', [BeritaController::class, 'insertBerita'])->name('insertDataBerita');
214 Route::post('/updateDataBerita', [BeritaController::class, 'updateBerita'])->name('updateDataBerita');
215
216 Route::get('/DataPortfolio', [PortfolioController::class, 'getAllPortfolio'])->name('DataPortfolio');
217 Route::post('/insertDataPortfolio', [PortfolioController::class, 'insertPortfolio'])->name('insertDataPortfolio');
218 Route::post('/updateDataPortfolio', [PortfolioController::class, 'updatePortfolio'])->name('updateDataPortfolio');
219
220 Route::get('/DataInfocudari', [InfocudariController::class, 'getAllInfocudari'])->name('DataInfocudari');
221 Route::post('/insertInfocudari', [InfocudariController::class, 'insertInfocudari'])->name('insertInfocudari');
222 Route::post('/updateInfocudari', [InfocudariController::class, 'updateInfocudari'])->name('updateInfocudari');
223

```

Figure 5. Structured Routing Implementation in Instructor Data Management

Structured routing has also been successfully applied to the training data management feature. For this feature, the routing code for the website pages is placed in the web.php file immediately below the routing code for the instructor data management feature. The code cannot be combined because this routing is specific to the training data management feature. The routing code for the training data management feature is shown in Figure 6 below.

```

200 Route::get('/DataMateri', [MateriController::class, 'getAllMateri'])->name('DataMateri');
201 Route::post('/insertDataMateri', [MateriController::class, 'insertMateri'])->name('insertDataMateri');
202 Route::post('/updateDataMateri', [MateriController::class, 'updateMateri'])->name('updateDataMateri');
203
204 Route::get('/DataKarir', [KarirController::class, 'getAllKarir'])->name('DataKarir');
205 Route::post('/insertDataKarir', [KarirController::class, 'insertKarir'])->name('insertDataKarir');
206 Route::post('/updateDataKarir', [KarirController::class, 'updateKarir'])->name('updateDataKarir');
207
208 Route::get('/DataTrainer', [TrainerController::class, 'getAllTrainers'])->name('DataTrainer');
209 Route::post('/insertDataTrainer', [TrainerController::class, 'insertTrainer'])->name('insertDataTrainer');
210 Route::post('/updateDataTrainer', [TrainerController::class, 'updateTrainer'])->name('updateDataTrainer');
211
212 Route::get('/DataBerita', [BeritaController::class, 'getAllBerita'])->name('DataBerita');
213 Route::post('/insertDataBerita', [BeritaController::class, 'insertBerita'])->name('insertDataBerita');
214 Route::post('/updateDataBerita', [BeritaController::class, 'updateBerita'])->name('updateDataBerita');
215
216 Route::get('/DataPortfolio', [PortfolioController::class, 'getAllPortfolio'])->name('DataPortfolio');
217 Route::post('/insertDataPortfolio', [PortfolioController::class, 'insertPortfolio'])->name('insertDataPortfolio');
218 Route::post('/updateDataPortfolio', [PortfolioController::class, 'updatePortfolio'])->name('updateDataPortfolio');
219
220 Route::get('/DataInfocudari', [InfocudariController::class, 'getAllInfocudari'])->name('DataInfocudari');
221 Route::post('/insertInfocudari', [InfocudariController::class, 'insertInfocudari'])->name('insertInfocudari');
222 Route::post('/updateInfocudari', [InfocudariController::class, 'updateInfocudari'])->name('updateInfocudari');
223

```

Figure 6. Structured Routing Implementation in Training Data Management

Application of Livewire

The Livewire technology has been successfully applied to the instructor data management feature. This feature displays all existing instructor data with just one line of code. This is because Livewire encapsulates both backend and frontend logic code for querying the database related to instructor data and the sub-view for displaying the query results. The Livewire code for the instructor data management feature is shown in Figure 7 below.

```

28 </a>
29 </div>
30 </div>
31 <div class="flex flex-row gap-2 items-center justify-center">
32 <a href="https://www.instagram.com/mncacademy/">
33 
34 </a>
35 <a href="https://www.facebook.com/mncacademy/">
36 
37 </a>
38 <a href="https://www.twitter.com/mncacademy/">
39 
40 </a>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>

```

Figure 7. Livewire Implementation in Training Data Management

The results of the Livewire implementation for the instructor data management feature can be seen in Figure 8 below.

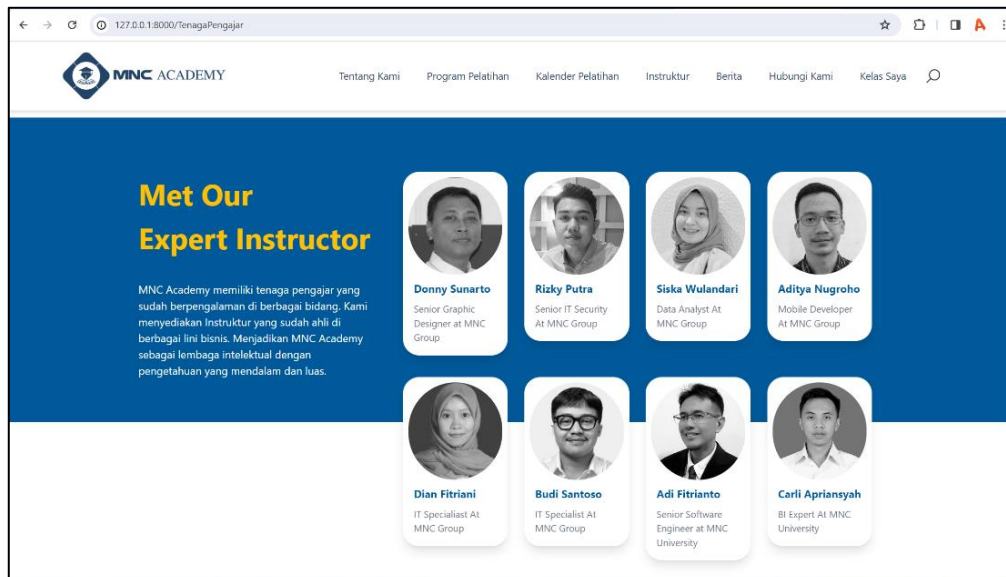


Figure 8. Feature of Instructor List

Livewire technology has also been successfully implemented in the training data management feature. In this section, Livewire code is placed in the home.blade.php page to display all existing training data with just one line of code. This simplifies the developer's task in building web pages with extensive content. Using Livewire, the logic code for querying all training data from the database, creating lists, populating lists, looping through the lists, retrieving parameters on the frontend, looping through the display of training data, and organizing the display is encapsulated into a single line of code, as shown in Figure 9 below.

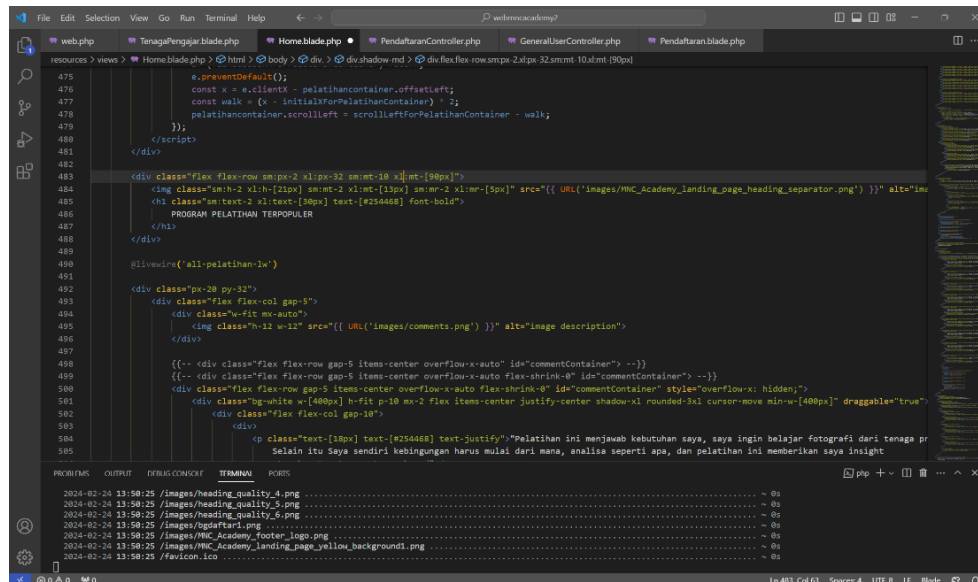


Figure 9. Livewire on Training List Feature

The result of the Livewire implementation for the training data management feature can be seen in Figure 10 below.

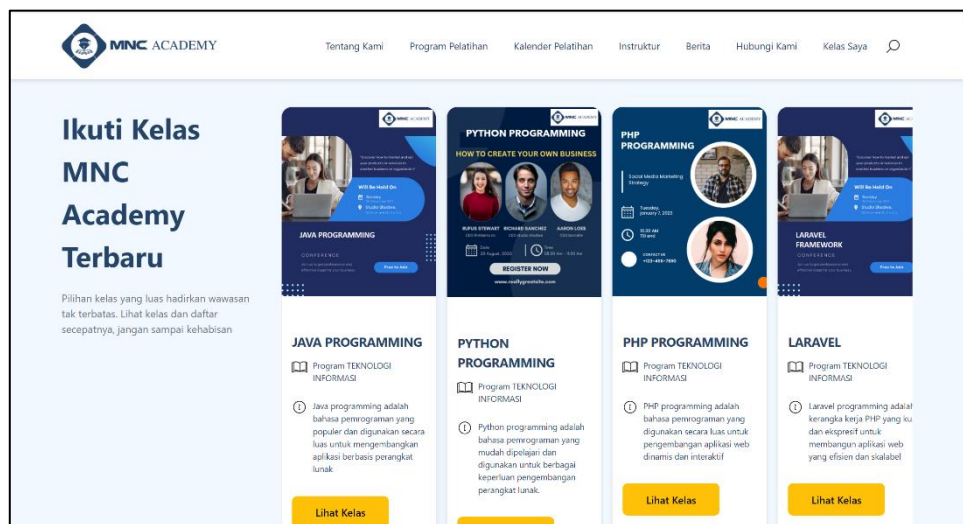


Figure 10. Feature of Training Data Management

DISCUSSION

The application of structured routing and Livewire technologies in this system has proven to be effective in organizing and managing data management features. Structured routing successfully organized the routing operations for the instructor and training data management features, ensuring modularity and clarity in the codebase. This modular approach enhances maintainability and simplifies debugging by isolating routing rules for each feature, as demonstrated in Figure 5. The separation of routing logic contributes to a more organized and scalable system compared to a single-file routing approach.

Livewire, on the other hand, streamlined the integration of frontend and backend logic, reducing code complexity and improving efficiency. The ability to manage data interactions with minimal code in both the instructor and training data management features demonstrates Livewire's capability to enhance developer productivity and performance. The effectiveness of Livewire is evident in its application for the instructor data management feature, as shown in Figure 7. Overall, these technologies significantly improve the system's efficiency and maintainability, providing valuable insights into effective web application development practices.

CONCLUSION

This research has produced an application with two main features: instructor data management and training data management. The application is developed using PHP programming language, Laravel framework, and MySQL for database management.

Based on the discussion above, it can be concluded that the application meets the expectations for managing extensive and complex instructor and training data. The integration of structured routing and Livewire technologies enhances the functionality of the application, benefiting both users and developers. However, further studies could explore additional features and optimizations to improve the application's scalability and performance. Future research may also focus on addressing any limitations identified during implementation, such as user interface enhancements or advanced data analytics capabilities

REFERENCES

- Agus, P., Shelve Nidya Neyman, & Hendra Rahmawan. (2023). Pengembangan sistem manajemen pelatihan kerja di Kota Surakarta. *Jurnal Teknologi Informasi Dan Ilmu Komputer/Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(4), 865–872.
- Ambriani, D., & Nurhidayat, A. I. (2020). Rancang bangun repository publikasi ilmiah dosen berbasis web menggunakan framework Laravel. *Jurnal Manajemen Informatika*, 10(1), 1–4. <https://doi.org/10.24844/mij.v10i1.10221>
- Daru, A. F., & Adhiwibowo, W. (2021). Penerapan metode rapid application development untuk mengembangkan sistem informasi stok barang menggunakan Livewire Laravel. *Jurnal Teknologi Informasi dan Komunikasi*, 12(2), 48–57.
- Firmansyah, A., & Zen, B. P. (2023). Company profile application system at PT. Powerkerto Wahyu Keprabon using Scrum method. *Journal of Innovation Information Technology and Application (JINITA)*, 5(1), 1–11.
- Hakiki, M., Fadli, R., Putra, Y., Pertiwi, I., Muhammadiyah, S., & Bungo, M. (2021). Perancangan sistem informasi manajemen berbasis sekolah SMA Negeri 1 Muara Bungo. *Jurnal Muara Pendidikan*, 6(1).
- Madani, P. O., Munti, N. Y. S., & Gusman, D. (2023). Sistem administrasi kerja praktik mahasiswa Fakultas Teknik Universitas Pahlawan Tuanku Tambusai berbasis website. *Jurnal Inovasi Teknik Informatika*, 8(1), 1–8.
- Novianto, R., & Maryam. (2022). Sistem informasi penjualan berbasis web pada UMKM R-DUA Lencana Kudus. *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, 10(2), 35–42.
- Pasaribu, Y. K., Nurhayati, O. D., & Windasari, I. P. (2022). Sistem informasi geografis lokasi kantor pemerintahan di kota Semarang berbasis web. *Jurnal Ilmu Teknik dan Komputer*, 6(1), 59–59.
- Rofiq, A., Sholiq, S., & Muqtadiroh, F. A. (2013). Rancang bangun aplikasi sistem informasi manajemen pelatihan (SIMPEL) pada Lembaga Pelatihan XYZ. *Jurnal Teknik ITS*, 2(1), 2977-2984. <https://doi.org/10.12962/j23373539.v2i1.2977>
- Sofyan, A., Sadikin, A., Sefriyandi, R., & Djutalov, R. (2023). Sistem informasi pengelolaan data guru dan siswa SMK Negeri 2 Tangerang Selatan berbasis web. *LOGIC : Jurnal Ilmu Komputer Dan Pendidikan*, 1(3), 460–466.
- Supriyanto, A., & Susanto, A. (2019). Rancang bangun aplikasi helpdesk berbasis web menggunakan framework Laravel (studi kasus di PDAM Surya Sembada Kota Surabaya). *Jurnal Manajemen Informatika*, 9(2), 123–130. <https://doi.org/10.24844/mij.v9i2.10082>
- Suhayati, E., Novianti, W., & Zakiah, A. N. (2023). Website-based catering service information system. *Journal of Engineering Science and Technology*, 17(6), 3956–3972. <https://repository.unikom.ac.id/70230/>
- Widya Ayu Suryaningtyas, Adam Hendra Brata, & Lutfi Fanani. (2020). Pengembangan sistem manajemen data siswa sekolah dasar berbasis web (studi kasus: SDN Wadungasri Sidoarjo). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(10), 3607–3613.
- Yudhistira, R., Setyawan, A., & Harsiti. (2022). Tracking system pengelolaan berita untuk jurnalis pada portal berita online. *ProTekInfo (Pengembangan Riset dan Observasi Teknik Informatika)*, 9(1), 1–7.
- Zy, A. T. (2022). Penerapan metode RAD untuk pengembangan sistem inventori obat berbasis web menggunakan framework Laravel (studi kasus pada klinik Sukaraya Medika Kabupaten Bekasi). *Pelita Teknologi*, 17(1), 12–28.