Final Project Information System Design in D3 Telecommunication Engineering with Laravel-Based Data Security Implementation

Dewi Vista Oktaviani Napitupulu[1], Dianthy Marya[2*], Muhammad Syirajuddin Suja'i[3]

[1,2] Digital Telecommunication Network Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

[3] Telecommunication Engineering Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

[1]dewinapitupuluu00@gmail.com , [2]dianthy@polinema.ac.id, [3]muh_syirajuddin@polinema.ac.id

***Abstract*— Final project management in D3 Telecommunication Engineering requires an integrated and secure system to ensure the smooth running of the final project management process and the protection of academic data. This study aims to design a web-based final project information system using the Laravel framework, as well as to identify and mitigate potential security vulnerabilities against brute force and SQL Injection attacks using the OWASP ZAP Fuzzer. The research method used is Research and Development (R&D). The results of blackbox testing on 23 test scenarios, all of which were declared successful, indicating that each system feature has run according to its expected function. The scan results show 31 potential vulnerabilities, consisting of 1 high risk, 4 medium, 6 low, and 20 informational alerts. Brute Force testing was carried out for 51 login attempts, while SQL Injection for 17 payload attempts. The test results show that all requests result in HTTP 302 (Found) which indicates incorrect credentials, the system cannot be penetrated, and does not trigger database errors or changes in data display. The response header size of 1220 bytes and the response body size of 386 bytes indicate that the server consistently returns the same failed page without any additional error messages, and the entire payload is deemed harmless by the server. Thus, the system is proven secure against both attacks and capable of supporting the integrated final project management process.**

***Keywords*— Brute Force, Information Systems, Final Project, Laravel, OWASP ZAP, SQL Injection.**

I. INTRODUCTION

A final assignment is a requirement for students in their final semester as a requirement for graduation from college, including Diploma 3 Telecommunication Engineering students at Malang State Polytechnic. The final assignment involves several stages, including registration for the proposal seminar and final assessment. Currently, all of these processes are still carried out manually, including the physical submission of complete registration documents to the committee and the in-person guidance logbook. This manual process involves numerous documents, data, and interactions between students, supervisors, the committee, and other relevant parties. Disadvantages of these processes include the need for large amounts of storage space, especially if the number is large, the potential for data loss, and limited access if the supervisor is unable to attend. Furthermore, the current registration process uses a Google Form created by the committee for each registration opening, requiring time to create the form. The committee then exports the Google Form summary results using Excel and then reorganizes them using Microsoft Word to conform to the specified document format. The summarized data compiled in this document is then shared via WhatsApp with student groups. This process is time-consuming, repetitive, and prone to data input errors. Guidance activities have long been crucial between lecturers and students. Each supervisor has the right to approve the progress of a student's final assignment, from chapter to chapter, and the overall discussion. Various obstacles, such as busy teaching schedules, absences due to off-campus activities or illness, and the distance from home to campus, hinder guidance activities. This situation highlights the need for an integrated information system to support the management of final assignments and ensure proper documentation. Along with technological advancements, a final assignment information system is increasingly needed to simplify final assignment management [1]. This system enables organized data storage, facilitates information access, and accelerates the final assignment completion process.

Currently, almost all aspects of life depend on digital systems, including education. These systems can store data in an organized manner, facilitate information access, and accelerate the final assignment completion process. The increasingly widespread use of the internet allows access to real-time information and remote connections. However, despite the convenience offered, technological developments also bring new challenges, namely increasing cybersecurity threats. One of the most common threats is brute force attacks and SQL injection. Indonesia ranks second in the top brute force attacks in Southeast Asia after Vietnam, which ranks first. Between January and December 2023, a total of 61,374,948 Bruteforce.Generic.RDP.* attacks were detected and thwarted by Kaspersky B2B products [2]. Brute force attacks aim to

guess passwords by trying various character combinations. Furthermore, according to Haikal Muhammad, many types of web attacks are currently occurring. One frequently used by attackers is SQL Injection [3]. SQL Injection attacks exploit security vulnerabilities in web applications to manipulate databases. These attacks can lead to data leaks, system failures, and even the loss of critical data. This situation demonstrates that security aspects are a crucial part of designing information systems, not just focusing on functionality.

Previously, relevant research was conducted by Siti Qotijah and Kholifah Murniati, entitled "Development of a Web-Based Final Project Information System for the TRPL Study Program, Vocational High School, UGM." This research focused solely on system development and utilized the CodeIgniter framework [4]. However, this research did not include security testing or attack mitigation, particularly brute force attacks and SQL injection. Therefore, further research is needed to examine security aspects and mitigate potential cyberattacks on the final project information system. In this research, Laravel was chosen as the framework for developing the final project information system. Laravel is a popular PHP framework widely used by web developers today [5].

Previous studies generally focused on developing final project information systems using waterfall or prototype methods, with black-box functional testing and user assessment. Mustianti et al. (2020) developed a Laravel-based system with good black-box test results and a high level of system feasibility based on MOS. The results of the MOS questionnaire test showed that 95.34% of informatics student respondents, 99.33% of non-informatics student respondents, and 100% of lecturer and admin respondents agreed and the system was suitable for use [6]. Fatman et al.'s (2024) research used NextJS and PostgreSQL, following all waterfall stages and testing the system with black-box [7]. Saputra et al. (2022) developed a final project guidance system using a four-stage waterfall method with an average user satisfaction score of 4.5 [8]. Meanwhile, Renaningtias and Apriliani (2021) implemented a prototype model using UML to address the shortcomings of the previous system [9]. Unlike these studies, which emphasized functional development and testing, this study not only built a Laravel-based final project information system but also emphasized data security with protection against brute force and SQL injection to ensure the integrity and confidentiality of academic data.

According to [9] an information system is a component consisting of humans, information technology and work procedures that process, store, analyze and disseminate information to achieve a goal. Its function is to collect, store, process, and convey information with the aim of supporting decisions, planning, management, and other business or organizational activities.

Bootstrap is a front-end framework that helps speed up and simplify web and mobile development, particularly responsive web development. This responsive functionality adapts to the screen size of any browser used on a desktop or smartphone. Bootstrap provides a variety of useful features, including built-in HTML, CSS, and JavaScript, that are ready to use and easy to develop [10].

A Data Flow Diagram (DFD) is a logical model of data or processes created to describe where data originates and where data exits the system, where the data is stored, what processes produce the data, and the interactions between the stored data and the processes applied to it. A DFD depicts data storage and the processes that transform the data. A DFD shows the relationship between data in the system and processes in the system [11].

According to Gordon C. Everest in Didik Setiawan's book (2017) in the journal (Heriyanto, 2018): "A database is a collection of many interrelated data collected in one place and used by a centrally controlled application system and also has valuable value for the owner" [12]. According to Winarno and Utomo in (Prayitno & Safitri, 2015) in the journal (Sutrisno et al., 2020) it is explained that a database or can also be called a database which is a collection of interrelated data.

Visual Studio Code is a free and open source code editor developed by Microsoft. It is available for Windows, macOS, Linux, and can even be run in a web browser [13]. VSCode is a text editor platform that can run various programming languages Javascript, Typescript, and Node.js, as well as other programming languages with the help of plugins that can be installed via the Visual Studio Code marketplace such as: C++, C#, Python, Go, Java, PHP and others. VSCode is also equipped with various features that make it easier for developers to write script code.

Laravel is a PHP-based web framework created by Taylor Otwell as an open source and free [14]. The syntax of the Laravel programming framework is very beautiful and expressive. Laravel follows the model-view-controller (MVC) architectural pattern, which can help speed up the process of building web applications. MVC is a software approach that separates application logic from presentation. MVC separates applications based on their components, such as data manipulation, controllers, and user interfaces.

- Model: A model represents a data structure. Typically, a model contains functions that assist someone in database management, such as inserting data into the database, updating data, and so on.
- View: The view is the part that controls the display to the user. It can be likened to a web page.
- Controller: The controller is the part that bridges the model and view.

In addition, the advantages and disadvantages of Laravel are mainly related to security features, namely [15]:

a. Advantages

- CSRF (Cross-Site Request Forgery) Protection Feature: Laravel automatically adds a CSRF token to forms to prevent attacks that mimic user requests.
- Encryption Features: Laravel provides various data encryption methods, such as bcrypt for password encryption, ensuring more secure sensitive data. To encrypt data, use the encrypt() function, while to decrypt data, use the decrypt() function.

- Input Validation Features: Laravel has a robust input validation system, helping prevent injection attacks such as SQL Injection and XSS.
- Authentication Features: Laravel offers various authentication mechanisms, including password-based, token-based, and social media authentication.
- Authorization Features: Laravel provides a customizable authorization system to manage user access rights to various parts of the application.

b. Disadvantages

- Configuration Dependence: If the available security features are not configured correctly, the website remains vulnerable to cyberattacks.
- Regular maintenance following updates: developers need to adjust the code especially if the application being developed is quite complex.

Brute force is an attack technique or hacker action that forces a web security system by trying to guess usernames and passwords. Hackers use brute force attacks to find hidden code and web page vulnerabilities that can be exploited. Once identified, the attacker uses the information to infiltrate the system and compromise data. The ultimate goal of this attack is to cause a denial of service on a web page and extract data from the system to be directed to a third party [16].

One of the common information security threats on websites is SQL Injection [17]. SQL Injection is a method of injection and database security abuse that is used to intentionally insert a parameter on a website or a query statement with the aim of obtaining user data [18]. This attack can allow attackers to retrieve, modify, or even delete data from the database associated with the web application. SQL Injection vulnerabilities are usually caused by web applications that do not validate user input properly [19].

Open Web Application Security Project Zed Attack Proxy (OWASP ZAP) is a penetration testing tool developed by OWASP (Open Web Application Security Project) which focuses on improving software security [20].

One of the features is Fuzzing, which is the process of automatically sending a large amount of test data (payload) into the application's input parameters to observe the server's response. OWASP ZAP Fuzzing will send a large amount of test data (payload) to the input parameters on a web application [21]. All server responses are then recorded and analyzed, including HTTP status codes, response size, content, and response time.

II. METHOD

A. Type of Research

This research is a research and development (R&D) study program that aims to design and develop a Final Project Information System for the D3 Telecommunication Engineering study program. The research stages include needs analysis, system design, implementation, and testing.

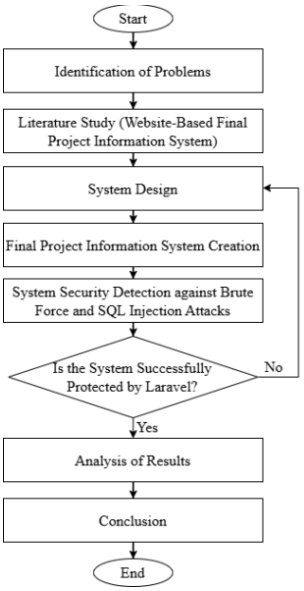

Figure 1. Flowchart Stages Research

Figure 1 illustrates the research flowchart as follows:

1. Problem Identification

   One of the research stages is problem identification, which involves identifying issues related to the need for a final project information system to manage the final project for the Diploma 3 Telecommunications Engineering program.

2. Literature Review

   After identifying the problem, the next step is to collect relevant sources such as books, journal articles, papers, and other documents related to the topic of designing a website-based final project information system.

3. System Design

   System design is the process of describing the structure and operation of the system through data flow diagrams and entity relationship diagrams. It also involves determining the software and hardware to be used in developing and testing the system.

4. System Development

   This stage involves creating an attractive, user-friendly interface and features that meet user needs.

5. System Security Detection against Brute Force and SQL Injection Attacks

   Security is crucial for website systems, especially educational websites. This stage aims to detect security vulnerabilities in the system to ensure the system is protected from cyberattacks, especially brute force and SQL injection attacks. Detection was performed using the open-source software OWASP ZAP.

6. Analysis of Results and Conclusions

   The results of the OWASP ZAP scan were then analyzed for security vulnerabilities against brute-force attacks and SQL injection. If these vulnerabilities were identified, mitigation measures were then identified to reduce or prevent both attacks. The mitigation measures taken were related to the security systems available in Laravel.

B. Data Flow Diagram

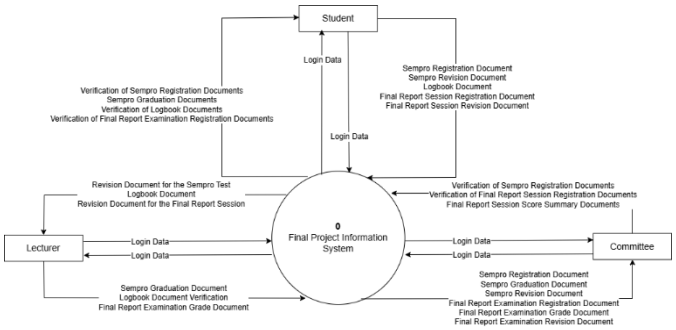

Figure 2. DFD Level 0 System Process

Figure 2 illustrates a context diagram or data flow diagram (DFD) at level 0. Students can upload required documents for the defense, including proposals and final projects, and complete the logbook. The committee has the authority to verify the registration documents. Lecturers can determine the graduation status of proposal seminars, verify the logbook, and assign final project defense grades to students.

C. Database Table Design

The database table design shows the data structure in the database so that the data in the database can be stored in a structured manner and can run optimally, as shown in Table I.

TABLE I
PROPOSAL TABLE

Nama Kolom	Tipe	Keterangan
id	bigint	Primary Key
prodi_id	unsigned bigint	Foreign Key
periode_id	unsigned bigint	Foreign Key
pendaftaran_sempro_id	unsigned bigint	Foreign Key
tahap_id	unsigned bigint	Foreign Key
bidang_minat_id	unsigned bigint	Foreign Key
jenis_judul_id	unsigned bigint	Foreign Key
status_sempro_proposal_id	unsigned bigint	Foreign Key
status_sempro_penguji_1_id	unsigned bigint	Foreign Key
status_sempro_penguji_2_id	unsigned bigint	Foreign Key
penguji_sempro_1_id	unsigned bigint	Foreign Key
penguji_sempro_2_id	unsigned bigint	Foreign Key
penguji_sidang_ta_1_id	unsigned bigint	Foreign Key
penguji_ sidang_ta _2_id	unsigned bigint	Foreign Key
dosen_pembimbing_1_id	unsigned bigint	Foreign Key
dosen_pembimbing_2_id	unsigned bigint	Foreign Key
pendaftaran_semhas_id	unsigned bigint	Foreign Key
periode_semhas_id	unsigned bigint	Foreign Key
tahap_semhas_id	unsigned bigint	Foreign Key
status_semhas_proposal_id	unsigned bigint	Foreign Key
status_semhas_penguji_1_id	unsigned bigint	Foreign Key
status_semhas_penguji_2_id	unsigned bigint	Foreign Key
judul	varchar (255)	
topik	varchar (255)	
tujuan	varchar (1000)	
latar_belakang	text	
blok_diagram_sistem	varchar (255)	
created_at	timestamp	
updated_at	timestamp	

III. RESULTS AND DISCUSSION

The following are the results of the implementation of the software interface and functionality.

A. Design and Implementation Results

1. Login Page

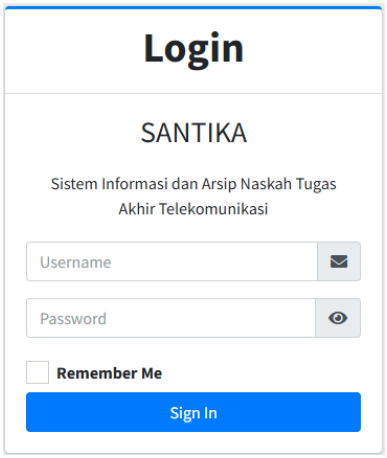

Figure 3. Login Page

The figure 3 is login page displays a form containing a username and password. The student username is the Student ID Number (NIM), while the lecturer's username is the National ID Number (NIDN). Additionally, there's a "Remember Me" option, a checkbox, that allows the system to save the user's login session. At the bottom, there's a blue "Sign In" button that sends the login data to the system.

2. Website Design Results on Admin

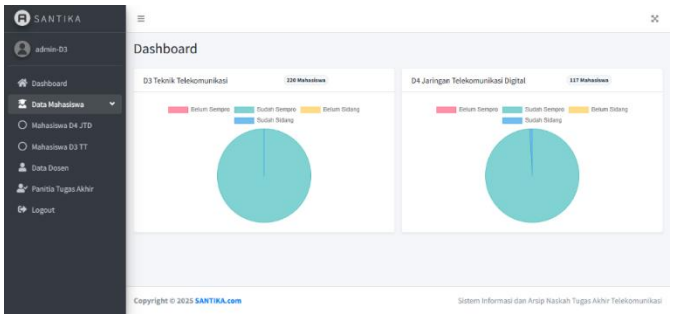

Figure 4. Admin Page

The main page (right) displays two pie charts, each of which is divided into several color categories that indicate the student's status, namely "not yet at the proposal seminar", "already at the proposal seminar", "not yet at the exam", and "already at the exam". Admins are responsible for adding student, lecturer, and final project committee data. Uploaded data must be in .xlsx spreadsheet format.

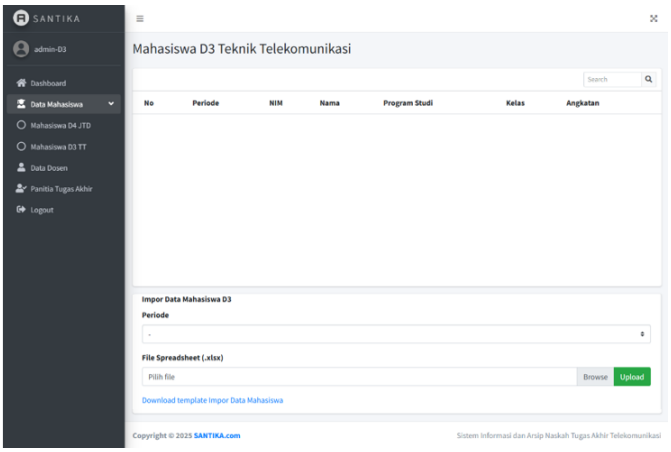

Figure 4. Student Data Upload Menu Page in Admin

Figure 5 the D3 Student Data Import feature allows administrators to bulk upload student data via .xlsx spreadsheet files. Admins can select academic periods, upload files, and download import templates to ensure the data format matches the system. The student data table for the Diploma 3 Telecommunication Engineering program contains student information such as student ID number (NIM), name, study program, class, and intake. The lecturer data table contains student ID number (NIDN), student ID number (NIP), and lecturer name.

3. Website Design Results for Students

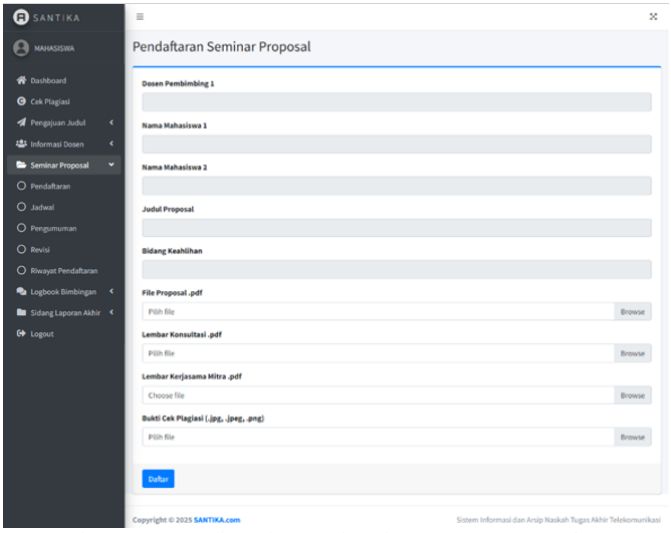

Figure 5. Proposal Seminar Registration Menu Page at Student

When registering for the proposal seminar, students must fill out an input form and upload documents. This form includes several information fields, such as the first supervisor, the name of student 1, the name of student 2, the proposal title, and the field of expertise. After that, students are required to upload supporting files, including the proposal file (.pdf), consultation sheet (.pdf), an optional partner collaboration sheet (.pdf), and proof of plagiarism check (.jpg/.jpeg/.png).

4. Website Design Results for Lecturers

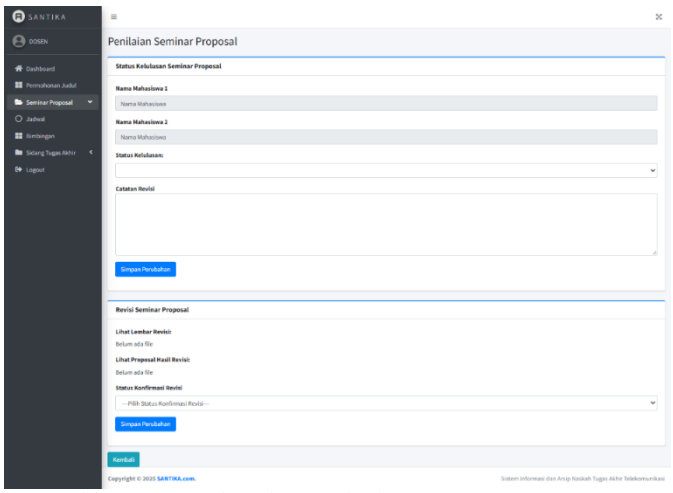

Figure 6. Proposal Seminar Graduation Status Page at Lecturer

Figure 6 shows the seminar proposal assessment page used by lecturers to assign grades and revision notes. The first section contains the Graduation Status form, including the student's name, status options (pass, revised, or failed), and a Revision Notes column. The lecturer then saves the evaluation using the "Save Changes" button.

The second section is the Seminar Proposal Revisions, which is used to view and assess revised documents using the "View Revision Sheet" and "View Revised Proposal" features. Lecturers can also set the Revision Confirmation Status via the dropdown menu and save updates using the "Save Changes" button.Website Design Results for Committee

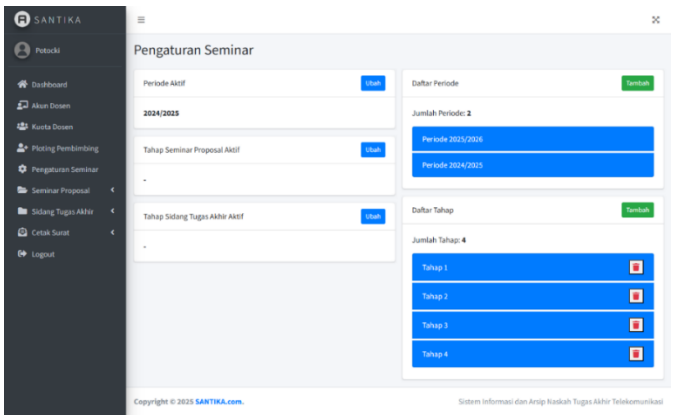

Figure 7. Seminar and Session Settings Menu Page in the Committee

Figure 7 above shows that on the main page, users can set the active period, proposal seminar stage, and final project hearing stage using the "Change" button. Meanwhile, on the right side, there's a list of seminar periods and stages that can be added or removed using the "Add" button and the delete icon. This view makes it easier for users to organize the seminar schedule and stages in a structured manner.

B. Functional Testing

The following are the results of testing the functionality of the website used by various roles, which are shown in the table II.

TABLE 1I
BLACK BOX TESTING RESULTS

Scenario	Input	Output	Information
Login	Login with the appropriate username and password pair	Users can enter the main page, namely the dashboard, after logging in.	Success
Add data	Add data for lecturers, students and final project committee	Admin successfully saved data on lecturers and students as well as the chairperson and members of the D3 thesis implementation team.	Success
Proposal Seminar	Uploading proposal seminar registration documents	Students successfully saved all documents and submitted their proposal seminar registration.	Success
	Verifying student registration documents for proposal seminar	Committee has successfully verified the proposal seminar registration document.	Success
Logbook	Filling out the logbook	Students successfully submitted logbook verification request.	Success
	Verifying student guidance logbook	Lecturer successfully views the complete list of students being mentored.	Success
Scenario	**Input**	**Output**	**Information**
Final Session	Uploading final trial registration documents	Students successfully saved all documents and submitted the final project defense registration.	Success
	Verifying student registration documents for final project defense	Committee has successfully verified the final project trial registration documents.	Success
	Filling in the final exam scores	Lecturer successfully confirmed the request for verification of the final assignment revision document.	Success
Logout	Logout	The system can stop all processes and return to the initial login page.	Success

Blackbox testing was conducted to ensure each feature functioned according to specifications, and the results in table 1 show successful test scenarios, indicating the system has met the functional requirements of the Diploma 3 Telecommunication Engineering Study Program. The Laravel-based system uses an MVC architecture, where the Model manages data and relationships between tables, the View provides a user-friendly interface through a Blade Template and is protected by CSRF tokens, and the Controller handles application logic such as authentication, authorization, input validation, and middleware settings for role-based access control. Role division allows students to only upload documents and monitor guidance, lecturers to provide approval or comments, committees to manage schedules and assessments, and admins to manage lecturer, student, and committee data, thus minimizing unauthorized access. The system is also designed modularly, consisting of authentication modules, guidance logbooks, assessments, and others, so that development and improvements can be made without changing the entire system. Based on testing on the admin, student, lecturer, and committee roles, all functions run well, the display is responsive, navigation is smooth, and data can be processed accurately. Thus, the system is declared ready for implementation in a real academic environment.

Based on the results of a vulnerability scanner on the target website santikapolinema.my.id using OWASP ZAP, 31 vulnerabilities were found. These vulnerabilities consisted of 1 high alert, 4 medium alerts, 6 low alerts, and 20 informational alerts. The overall results of the vulnerability scanner performed on the target website santikapolinema.my.id are presented in Table III.

TABLE III
SCANNING RESULTS ON WEBSITES USING OWASP ZAP

No.	Risk Warning	Type
1	Vulnerable JS Library	High
2	Content Security Policy (CSP) Header Not Set	Medium
3	Cross-Domain Misconfiguration	Medium
4	Missing Anti-clickjacking Header	Medium
5	Vulnerable JS Library	Medium
6	Big Redirect Detected (Potential Sensitive Information Leak)	Low
No.	**Risk Warning**	**Type**
7	Cookie No HttpOnly Flag	Low
8	Cross-Domain JavaScript Source File Inclusion	Low
9	Strict-Transport-Security Header Not Set	Low
10	Timestamp Disclosure - Unix	Low
11	X-Content-Type-Options Header Missing	Low
12	Authentication Request Identified	Infomasional
13	Information Disclosure - Suspicious Comments	Infomasional
14	Modern Web Application	Infomasional
15	Re-examine Cache-control Directives	Infomasional
16	Retrieved from Cache	Infomasional
17	Session Management Response Identified	Infomasional
18	Tech Detected - Bootstrap	Infomasional
19	Tech Detected - Chart.js	Infomasional
20	Tech Detected - Cloudflare	Infomasional
21	Tech Detected - Font Awesome	Infomasional
22	Tech Detected - Google Font API	Infomasional
23	Tech Detected - Google Hosted Libraries	Infomasional
24	Tech Detected - HSTS	Infomasional
25	Tech Detected - HTTP/3	Infomasional
26	Tech Detected - LiteSpeed	Infomasional
27	Tech Detected - Moment.js	Infomasional
28	Tech Detected - Select2	Infomasional
29	Tech Detected - jQuery	Infomasional
30	Tech Detected - jQuery UI	Infomasional
31	Tech Detected - jsDelivr	Infomasional

Brute force and SQL Injection testing was carried out using the Fuzzer feature on OWASP ZAP targeting the login page.

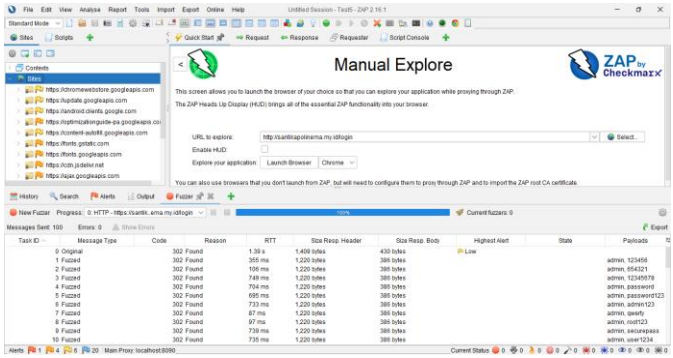

Figure 7. Fuzzing Brute Force Attack Process

Brute force testing using OWASP ZAP Fuzzer on the login endpoint (POST /login) was conducted with a wordlist of usernames and passwords. Of the 51 attempts, all requests resulted in an HTTP 302 Found error, indicating incorrect credentials, and the user was redirected to the login page. The request header size remained at 1574 ms because the header configuration (user-agent, content-type, cookie) remained unchanged. The response body size was also consistent at 386 bytes because the system always displayed the login failure page with the same message. The response header was also stable at 1220 ms because the server sent the same metadata on each attempt without dynamic changes. Meanwhile, the request body size varied between 78–91 ms depending on the length of the sent payload. Overall, the test results showed no indication of a brute force payload successfully breaching the login.

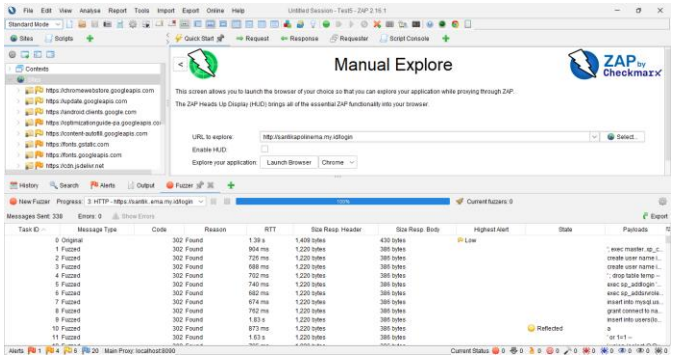

Figure 8. SQL Injection Fuzzing Process

SQL Injection testing with OWASP ZAP on the login endpoint (POST /login) using 17 SQLi payloads (error-based, boolean, blind, and union) showed that all requests resulted in HTTP 302 Found, indicating that all attempts were redirected back to the login page and that no payloads affected the application flow or queries on the server. The request header size was consistent at around 1574 bytes because the client metadata did not change, while the response body was stable at 386 bytes, indicating that the server always returned a failed login page without an SQL error message or data output. The response header was also consistent at 1220 bytes because the header content was the same for each request. Variation was only seen in the request body, which ranged from 79–127 bytes following the payload length. RTT times ranged from 660–750 ms, with a peak value of 3930 ms on the *x' and email is NULL* payload, likely due to network latency or longer server processing. Based on the results in table 4.7, no indication of SQL Injection was found; all payloads are rejected or treated as normal input without triggering errors or data exfiltration.

IV. CONCLUSION

The Final Project Information System for the Diploma 3 Telecommunication Engineering program was successfully built by implementing Laravel-based data security, the MVC concept, and role-based access control authentication. Functional testing results using 23 blackbox testing scenarios showed that all functions functioned as expected. Furthermore, each user, including admins, lecturers, committee members, and students, could only access features according to their respective access rights. Testing with the OWASP ZAP Fuzzer identified 31 potential vulnerabilities (1 high, 4 medium, 6 low, and 20 informational). Brute Force testing (51 attempts) and SQL Injection testing (17 payloads) all resulted in HTTP 302 (Found) errors indicating invalid credentials, with a response header of 1220 bytes and a response body of 386 bytes. Consequently, the server consistently returned the same failed page without any database errors or visual changes. All payloads were deemed harmless by the server, and no evidence of exploitation was found. Overall, the final project information system for the Diploma 3 Telecommunication Engineering program utilized Laravel security features such as input validation, password hashing (bcrypt), authentication middleware, CSRF protection, and login rate limiting. These features effectively prevented brute force attacks and SQL injection. Test results showed no malicious payloads that could modify data or alter application logic, with server responses remaining consistent.

REFERENCES

[1] F. S. Suwita, “Pengembangan Sistem Informasi Tugas Akhir dan Skripsi (SIMITA) di Universitas Komputer Indonesia(UNIKOM),” *Jurnal Teknologi dan Informasi (JATI),* vol. 10, pp. 71--82, 2020.

[2] G. Patoni, Y. Muhyidin dan D. Singasatia, “Implementasi WazuhPada UbuntuServerUntuk Mendeteksi Serangan Brute Force Hydra,” *Merkurius : Jurnal Riset Sistem Informasi dan Teknik Informatika,* vol. 2, pp. 145-156, 2024.

[3] M. Fadillah Dan Y. Servanda, “Analisis Efektivitas Teknik Parameterized Queries Dalam Mencegah Serangan Sql Injection Menggunakan Dvwa,” *Jupiter Teknologi Informatika & Komputer,* Vol. 5, Pp. 57-69, 2024.

[4] S. Qotijah Dan K. Murniati, “Pengembangan Sistem Informasi Tugas Akhir Berbasis Web Programstudi Trpl Sv Ugm,” *Journal Of Internet And Software Engineering (Jise),* Vol. 5, Pp. 37-42, 2024.

[5] L. Rahmawati Dan S. , “Desain Pengembangan Website Dengan Arsitektur Model View Controller Pada Framework Laravel,” *Jurnal Teknologi Dan Sistem Informasi Bisnis,* Vol. 6, Pp. 785-790, 2024.

[6] M. I. B. K. Widiartha Dan M. A. Albar, "Sistem Informasi Tugas Akhir Program Studiteknikinformatika Universitas Mataram," *Jtika,* Vol. 2, Pp. 19-29, 2020.

[7] Y. Fatman, I. S. Hidayat Dan N. I. Anadhiya, "Rancang Bangun Sistem Informasitugas Akhir Menggunakan Metode Waterfall Di Fakultas Teknik Universitas Islam Nusantara," *Edusaintek: Jurnal Pendidikan, Sains Dan Teknologi,* Vol. 11, Pp. 290-299, 2024.

[8] D. Saputra, H. A. Surniandari, M. Dan F. Akbar, "Sistem Informasi Bimbingan Tugas Akhir Mahasiswa Berbasis Website Menggunakan Metode Waterfall," *Matrik: Jurnal Manajemen, Teknik Informatika, Dan Rekayasa Komputer,* Vol. 21, Pp. 403-416, 2022.

[9] D. Anjeli, S. T. Faulina Dan A. Fakih, "Sistem Informasi Perpustakaan Sekolah Dasar Negeri 49 Oku Menggunakan Embarcadero Xe2 Berbasis Client Server," *Jurnal Informatika Dan Komputer (Jik),* Vol. 13, Pp. 57-66, 2022.

[10] T. J. Riasinir Dan W. , "Pemanfaatan Framework Bootstrap Dalam Merancang Website Responsif Untuk Toko D2 Adventure," *Jurnal Enter,* Vol. 2, Pp. 346-355, 2019.

[11] D. Triananda, M. Arafat Dan D. Pujianto, "Penggembangan Sistem Informasi Penggajian Karyawan Pada Smp Pgri 3 Baturaja Berbasis Web," *Jurnal Teknik Informatika Mahakarya (Jtim),* Vol. 6, Pp. 9-20, 2023.

[12] Y. Heriyanto, "Perancangan Sistewm Informasi Rental Mobil Berbasis Web Pada Pt.Apm Rent Car," *Jurnal Intra-Tech,* Vol. 2, Pp. 64-77, 2018.

[13] K. S. Ningsih, N. J. Aruan Dan A. T. A. A. Siahaan, "Aplikasi Buku Tamu Menggunakan Fitur Kamera Dan Ajax Berbasis Website Pada Kantor Dispora Kota Medan," *Sitek: Jurnal Sains, Informatika, Dan Tekonologi,* Vol. 1, Pp. 94-99, 2022.

[14] Y. Siyamto, J. Triyanto Dan M. R. Alwatoni, "Implementasi Framework Laravel Dalam Perancangan Sistem Informasi Penjualan Ukm Kota Surakarta," Dalam *Prosiding Seminar Nasional Ilmu Sosial Dan Teknologi 5 Tahun 2023*, Batam, 2023.

[15] S. M. Husain, L. Azhari, M. L. Aksani Dan S. A. Saputra, "Analisis Dan Implementasi Fitur Keamanan Aplikasi Pada Framework Laravel," *Jika (Jurnal Of Informatics),* Vol. 8, Pp. 281-287, 2024.

[16] R. A. Febrian, Y. Muhyidin Dan D. Singasatia, "Analisis Penyerangan Bruteforce Terhadap Secure Shell (Ssh) Menggunakan Metode Penetration Testing," *Jurnal Ilmiah Sain Dan Teknologi,* Vol. 2, Pp. 151-162, 2024.

[17] Y. Natanael, R. Felicia Dan E. M. S. Sakti, "Analisis Keamanan Informasi Bagi Pengguna Website Menggunakan Kalilinux Melalui Teknik Sql Injection," *Tekinfo,* Vol. 25, Pp. 123-132, 2024.

[18] F. A. Saputra, T. R. Dharmawan Dan A. Rustianto, "Implementasi Wazuh Siem Untuk Manajemen Log Event Di Pesantren Teknologi Informasi Dan Komunikasi Jombang," *Jurnal Informatikaterpadu,* Vol. 10, Pp. 146-155, 2024.

[19] F. Al Fajar, "Analisis Keamanan Aplikasi Web Prodi Teknik Informatika Uika Menggunakan Acunetix Web Vulnerability," *Jurnal Inova-Tif ,* Vol. 3, Pp. 110-120, 2020.

[20] N. F. Saragih, R. Tamalawe Dan I. M. Sarkis, "Analisis Dan Implementasi Secure Code Pada Pengembangan Sistem Keamanan Website Fikom-Methodist.Com Menggunakan Penetration Testing Dan Owasp Zap," *Jurnal Times,* Vol. 12, P. 28–39, 2023.

[21] B. Ariwibowo, "Analisis Keamanan Website Berbasis Black-Box Fuzzing: Studi Kasus Kerentanan Xss Dan Sql Injection Dalam Website X," *Jatikom: Jurnal Aplikasi Dan Teori Ilmu Komputer,* Vol. 7, Pp. 86-92, 2024.