

# SISTEM INFORMASI AKADEMIK POLINEMA

## PROJECT REPORT

POLITEKNIK NEGERI MALANG  
Jl. Soekarno Hatta no. 9, Malang, 65141  
polinema.ac.id

VERSION 1.0.0

01/12/2024

### VERSION HISTORY

VERSION	APPROVED BY	REVISION DATE	DESCRIPTION OF CHANGE	AUTHOR
1.0.0	Project Sponsor	Dec 02, 2024	Initial Version	Project Manager

PREPARED BY	TITLE	DATE
Pramana Yoga Saputra		01/12/2024

APPROVED BY	TITLE	DATE

## TABLE OF CONTENTS

1. PROJECT OVERVIEW .....	3
2. Objectives and Deliverables .....	3
3. Key Changes During Project Execution .....	3
4. PROJECT PERFORMANCE.....	4
5. Key Accomplishments.....	4
6. SYSTEM ARCHITECTURE .....	5
7. TECHNOLOGIES USED .....	5
8. APPLICATION FEATURES .....	6
9. USER INTERFACE DESIGN.....	6
10. TESTING AND QUALITY ASSURANCE .....	7
11. SECURITY MEASURES.....	8
12. CHALLENGES AND LESSONS LEARNED .....	8
13. FINANCIAL SUMMARY .....	9
14. STAKEHOLDER FEEDBACK .....	9
15. TRASNSITION AND HANDOVER .....	9
16. RECOMMENDATIONS.....	9

## 1. PROJECT OVERVIEW

The SIAKAD project aimed to create a centralized, web-based Academic Information System for Politeknik Negeri Malang. The project faced several changes during execution, including team member replacement, delays in integration, and budget adjustments. Despite these challenges, the project was successfully completed, delivering a system that met stakeholder expectations.

Project Name	SIAKAD Development Project
Start Date	September 2, 2024
End Date	December 1, 2024 (adjusted after delays)
Project Manager	Aditya Putra
Budget	Rp 51,000,000 (adjusted after hosting upgrade)
Status	Successfully Completed

## 2. OBJECTIVES AND DELIVERABLES

Objective	Status	Comments
Develop a responsive Academic Information System	Achieved	Fully functional on web and across devices.
Implement key functionalities for students, faculty, and administrators	Achieved	Features include course registration, grade management, and academic reporting.
Deliver secure authentication and data storage	Achieved	Implemented using Laravel framework and MySQL database.
Provide user training and documentation	Achieved	Training conducted; manuals delivered.

## 3. KEY CHANGES DURING PROJECT EXECUTION

- ❑ **Team Member Replacement:**
  - The Backend Developer resigned, and a replacement was onboarded.
  - **Impact:** A one-week delay in backend task completion during onboarding.
  - **Resolution:** Supervision and additional resources allocated to ensure continuity.
- ❑ **Delay in Integration Phase:**
  - Integration between frontend and backend components delayed by one week due to API mismatches.
  - **Impact:** Testing and deployment phases extended by one week.
  - **Resolution:** Teams collaborated closely to resolve API issues; updated schedule adhered to.

□ **Budget Adjustment:**

- Hosting plan upgraded to enhance performance and security, increasing costs by Rp 1,000,000.
- **Impact:** Budget increased from Rp 50,000,000 to Rp 51,000,000.
- **Resolution:** Cost absorbed through buffer allocation without affecting quality.

## 4. PROJECT PERFORMANCE

Metric	Planned	Actual
Timeline	September 2 - November 24, 2024	September 2 - December 1, 2024
Budget	Rp 50,000,000	Rp 51,000,000
Scope Completion	100%	100%
Key Risks Addressed	3 critical risks mitigated	No significant impact on deliverables.

## 5. KEY ACCOMPLISHMENTS

**Requirement Gathering and Design:**

Successfully gathered all stakeholder requirements and finalized UI/UX designs within the first month.

**Development:**

Both frontend and backend development completed, meeting all functional and security requirements.

**Testing and Deployment:**

System passed functionality, responsiveness, and security tests. User Acceptance Testing (UAT) was completed with positive feedback.

**Documentation and Training:**

Delivered comprehensive user and admin manuals; training sessions conducted for stakeholders.

## 6. SYSTEM ARCHITECTURE

The system architecture of SIAKAD follows a three-tier model, which ensures scalability, maintainability, and efficient separation of concerns:

### 6.1 Architecture Layers

1. **Presentation Layer (Frontend):**

- Technologies: HTML, CSS, JavaScript, and Bootstrap.
- Purpose: Provides a responsive user interface for students, faculty, and administrators to interact with the system.
- Features: Cross-browser compatibility and device responsiveness.

2. **Application Layer (Backend):**

- Technologies: Laravel Framework (PHP).
- Purpose: Handles server-side logic, API development, and business rules.
- Features: Authentication using Auth0 and RESTful API integration for seamless communication between components.

3. **Data Layer (Database):**

- Technologies: MySQL.
- Purpose: Securely stores academic data, including course details, grades, and user information.
- Features: Optimized queries for fast data retrieval and data integrity measures.

### 6.2 Data Flow

- User requests are sent from the frontend to the backend via secure APIs.
- The backend processes the request, accesses the database, and returns the required data to the frontend.

## 7. TECHNOLOGIES USED

Category	Technology	Purpose
Frontend Development	HTML, CSS, Bootstrap	Creates responsive and interactive user interfaces.
Backend Development	Laravel Framework	Manages server-side logic and APIs.
Database	MySQL	Stores user data, course records, and grades.

Hosting and Deployment	Niagahoster	Provides cloud hosting and domain registration.
Task Management	Trello	Tracks task assignments and project progress.
Performance Testing	Lighthouse	Evaluates system responsiveness and speed.
Security Testing	OWASP ZAP	Identifies and mitigates vulnerabilities.

## 8. APPLICATION FEATURES

### 1. **Course Registration:**

- Students can enroll in courses through an intuitive form.
- Faculty can manage course offerings via a centralized dashboard.

### 2. **Grade Management:**

- Faculty can input and modify student grades.
- Students can view grades in real-time.

### 3. **Academic Reporting:**

- Generates detailed reports for administrators on course enrollments, student performance, and departmental statistics.

### 4. **User Roles and Authentication:**

- Secure login system with role-based access for students, faculty, and administrators.
- Built using Auth0 for scalability and ease of management.

### 5. **Notifications:**

- Email alerts for deadlines, updates, and announcements using SendGrid's free tier.

## 9. USER INTERFACE DESIGN

### 9.1 Design Approach:

- **Wireframing Tools:** Figma was used to design user interface wireframes during the design phase.
- **Prototyping:** Mockups were reviewed and approved before development began.

### 9.2 Key Screens:

1. **Login Page:** Role-specific login with authentication prompts.
2. **Student Dashboard:** Displays enrolled courses, grades, and notifications.

3. **Faculty Dashboard:** Lists assigned courses, grading tools, and reporting features.
4. **Admin Panel:** Includes tools for managing user accounts, academic schedules, and generating reports.

### 9.3 Accessibility Features:

- Fully responsive design tested across mobile, tablet, and desktop devices.
- Clear navigation elements with consistent layouts.

## 10. TESTING AND QUALITY ASSURANCE

### 10.1 Testing Phases:

1. **Unit Testing:**
  - Tested individual components (e.g., login, course registration).
  - Ensured accurate functionality of isolated features.
2. **Integration Testing:**
  - Verified interaction between frontend and backend systems.
  - Ensured data flows correctly between API endpoints and the database.
3. **User Acceptance Testing (UAT):**
  - Conducted with a sample group of students, faculty, and administrators.
  - Positive feedback with minor issues resolved before deployment.

### 10.2 Testing Tools Used:

- **Lighthouse:** Evaluated system performance (response time and load speed).
- **OWASP ZAP:** Detected and resolved vulnerabilities in authentication and data flow.

### 10.3 Results:

Testing Type	Outcome
Unit Testing	All critical functions passed.
Integration Testing	Smooth interaction between components.
User Acceptance Testing	95% positive feedback. Minor bugs resolved.

## 11. SECURITY MEASURES

1. <b>Authentication:</b>
○ Role-based access implemented using Auth0, ensuring users only access authorized modules.
2. <b>Encryption:</b>
○ SSL/TLS encryption for secure communication between the client and server.
3. <b>Vulnerability Scanning:</b>
○ Conducted using OWASP ZAP to detect cross-site scripting (XSS) and SQL injection threats.
4. <b>Database Security:</b>
○ Utilized parameterized queries to prevent SQL injection.
○ Regular backups to ensure data recovery in case of failures.
5. <b>Testing and Monitoring:</b>
○ Continuous monitoring of system logs to detect anomalies.
○ Post-deployment audits scheduled monthly for the first quarter after release.

## 12. CHALLENGES AND LESSONS LEARNED

Challenge	Resolution
Replacement of Backend Developer	Onboarded replacement and allocated resources for supervision.
Integration issues between components	Dedicated additional time for integration testing and API alignment.
Budget Overrun	Absorbed additional hosting costs through buffer allocation.

**Lessons Learned:**

- Early stakeholder involvement is critical to reduce delays during validation phases.
- A dedicated testing phase significantly improves integration quality and reduces bugs.
- Allocating a buffer in budget and timeline helps mitigate unforeseen changes.



### 13. FINANCIAL SUMMARY

Cost Component	Planned Cost (Rp)	Actual Cost (Rp)	Variance
UI/UX Design	4,000,000	4,000,000	0
Frontend Development	15,000,000	15,000,000	0
Backend Development	15,000,000	15,000,000	0
Hosting and Deployment	575	1,575,000	+1,000,000
QA Testing	2,000,000	2,000,000	0
Miscellaneous	2,425,000	2,425,000	0
<b>Total</b>	<b>50,000,000</b>	<b>51,000,000</b>	<b>+1,000,000</b>

### 14. STAKEHOLDER FEEDBACK

**Academic Administrator:** "The system is faster and more reliable after hosting improvements."

**Head of Department:** "The adjustments were well-managed, and the project outcomes align perfectly with our goals."

**Students:** "The system is easy to use and makes course registration much simpler."

### 15. TRANSITION AND HANDOVER

- All system credentials and access details handed over to the IT department.
- Documentation and training manuals stored in the institutional repository.
- Post-deployment support will be provided for 1 month by the project team.

### 16. RECOMMENDATIONS

- Conduct periodic system updates to maintain performance and security.
- Monitor user feedback for potential enhancements in future iterations.

- Allocate additional buffer time for integration and testing in similar projects.

## APPROVAL

Name	Title	Signature
Aditya Putra	Project Manager	
Dr. Andi Wibowo	Head of Department	
Siti Rahmawati	Academic Administrator	