Implementation of the CodeIgniter Framework in the Development of a Cloud-Based Competency Assessment System

Made Pradnyana Ambara[1], Made Sudiarta[2], Sagung Mas Suryaniadi[3] and I Gede Teguh Satya Dharma[4]

[1,4] Information Technology Department, Politeknik Negeri Bali, Bali, Indonesia
[2] Tourism Department, Politeknik Negeri Bali, Bali, Indonesia
[3] Business Administration Department, Politeknik Negeri Bali, Bali, Indonesia
pradnyana_ambara@pnb.ac.id

Abstract. Competency assessment systems are essential for educational and professional certification frameworks, providing structured methodologies for evaluating knowledge, skills, and competencies. Traditional implementations, often based on localized servers and manual procedures, have limitations in scalability, prompt feedback, and efficient administration. This paper presents the development of a cloud-based competency assessment system using the CodeIgniter framework. The system features a modular architecture, real-time examination capabilities, automated scoring, and real-time analytics. Performance evaluations under realistic user load scenarios demonstrate the system's scalability, reliability, and efficiency. The results provide an architectural reference for integrating MVC frameworks with information systems and offer practical insights for developers aiming to build scalable, maintainable, and secure competency assessment applications.

Keywords: Cloud Computing, CodeIgniter Framework, Computational Assessment, Introduction

Competency assessment systems represent a critical element of educational and professional certification frameworks, offering structured methodologies for evaluating knowledge, skills, and competencies. Initial implementations were typically based on localized servers and manual procedures, which often proved inadequate in scaling to larger user bases, providing prompt feedback, and ensuring efficient administration (Nurjannah et al., 2022). The evolution of information systems has enabled centralized data management, flexible resource allocation, and widespread accessibility, effectively addressing many limitations of traditional architectures. This advancement allows institutions to deliver competency assessments with improved reliability and scalability (Guan et al., 2022).

Numerous prior studies have proposed information system-based platforms for competency assessment, yet these systems typically offer only basic functionalities such as user registration and document management, lacking support for real-time examination features (Deviana et al., 2023; Mustakim, 2024; Rafianto & Voutama,

A. A. N. G. Sapteka et al. (eds.), *Proceedings of the International Conference on Sustainable Green Tourism Applied Science - Engineering Applied Science 2025 (ICOSTAS-EAS 2025)*, Advances in Engineering Research 280,
https://doi.org/10.2991/978-94-6463-878-3_47

2025; Sabella et al., 2024; Sampurno et al., 2020; Sari et al., 2023; Rumini & Sulistiyani, 2025). Our research group has conducted multiple iterative studies aimed at enhancing system functionality (Ambara et al., 2023, 2024). The current work represents the latest improvement, introducing a fully integrated real-time examination module, marking a significant step toward a comprehensive and functional competency assessment platform using cloud computing. This paper examines whether CodeIgniter, when implemented within a modern information system architecture, can support dynamic test generation, automated scoring, and real-time analytics for competency assessment.

A modular system architecture was developed using CodeIgniter's core libraries and deployed in an environment configured for load balancing and auto-scaling. The system's performance was evaluated under realistic user load scenarios. The results provide an architectural reference for integrating MVC frameworks with information systems, present empirical performance metrics, including throughput and latency, and offer practical insights for developers aiming to build scalable, maintainable, and secure competency assessment applications.

1 Methodology

We conduct this research using the following steps: (1) Requirement Analysis; (2) Database Design; (3) User Interface Design; (4) Implementation; (5) Testing and Validation.

1.1 Requirement Analysis

The needs analysis aims to understand the system's objectives, the users it will serve, and the required functionalities. This stage involves activities such as observation, stakeholder interviews, and analysis of existing documents and business processes (Laplante & Kassab, 2022). These efforts ensure a comprehensive and contextually appropriate scope of requirements.

1.2 Database Design

After the system requirements are clear, the next stage is database design, which includes designing the conceptual model (ERD), logical model (normalization), and finally the physical database schema. The database design should reflect the relationships between entities and support query efficiency and data integrity. According to research by Galih et al. (2025), the application of normalization techniques up to the third normal form (3NF) has been proven to reduce data redundancy and improve query performance in financial risk management systems (Rafianto & Voutama, 2025).

1.3 Testing and Evaluation

The system's reliability and performance were validated through a structured testing approach, encompassing unit and integration testing to ensure the correctness of individual components and their interactions. To address scalability concerns, throughput testing was employed to evaluate the system's capacity to handle concurrent assessments, a critical requirement given the potential for multiple exams to occur simultaneously. Additionally, blackbox testing was utilized to assess functional and user interface (UI) behaviors from an end-user perspective, ensuring alignment with specified requirements and usability standards. These testing methodologies collectively ensured the system's robustness, efficiency, and adherence to certification process demands. Throughput, denoted as λ, quantifies the rate at which the system processes tasks or transactions within a specified time interval t, is defined as:

$$\lambda = \frac{N}{t} \tag{1}$$

Where N represents the total number of tasks. In the context of concurrent assessment systems, throughput is often expressed in requests per second (RPS) or transactions per second (TPS), depending on the operational metric (Alnuhait et al., 2024). To evaluate scalability under load, the average throughput $\hat{\lambda}$ is calculated as:

$$\hat{\lambda} = \frac{N_{total}}{t_{total}} \tag{2}$$

For peak performance analysis, the maximum throughput is determined by:

$$\lambda_{peak} = \frac{N_{peak}}{t_{peak}} \tag{3}$$

These metrics ensure the system's ability to handle simultaneous assessments, critical for maintaining service-level agreements in certification platforms. By monitoring key performance indicators such as response time, throughput, and error rates, the system can dynamically adjust resources to meet the demands of concurrent users.

2 Results and Discussion

2.1 Results

Entity-Relationship Diagram (ERD). The Entity-Relationship Diagram (ERD) illustrates a structured framework for managing certification assessment processes, emphasizing the interrelations among key entities involved in evaluating candidate competencies. The diagram encapsulates the core components of a system designed to organize assessments, track candidate progress, and align evaluations with defined certification schemes.

To ensure secure access and protection of sensitive information within the cloud-based competency assessment system, user authentication is implemented using modern encryption standards such as AES-256. Role-Based Access Control (RBAC) governs user permissions based on designated roles, enhancing system integrity. Additionally, system data is backed up weekly to external storage to support recovery and maintain operational continuity.

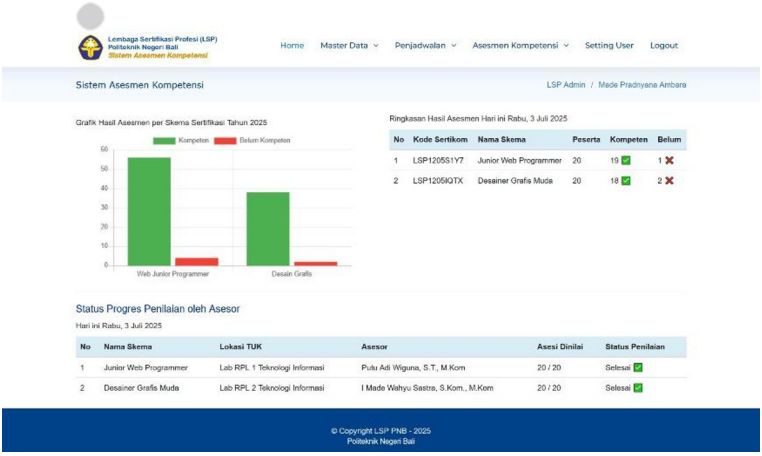

Figure 1. Dashboard Interface (in Indonesian)

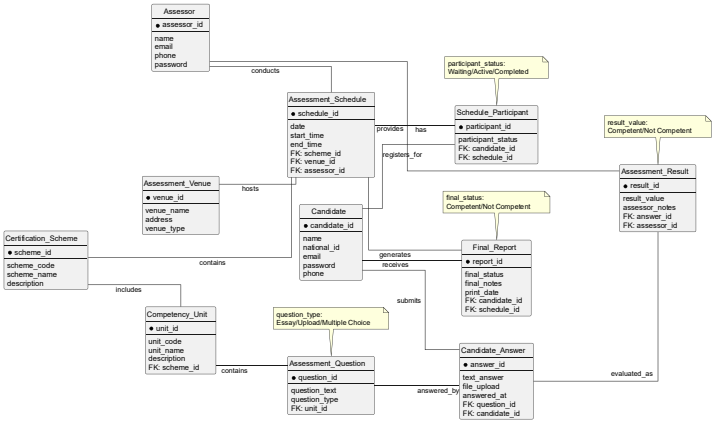

Figure 2. Entity Relationship Diagram

The Assessor entity comprises attributes such as assessor_id, name, contact details, and authentication credentials, enabling the identification and management of individuals responsible for evaluations. Similarly, the Candidate entity includes identifiers like candidate_id, personal information, and contact data, facilitating the tracking of participants in the certification process. The Certification_Scheme entity defines structured programs through scheme_id, scheme_code, and related descriptors, providing a framework for organizing assessment criteria.

Interactions between entities are governed by relationships that reflect operational requirements. For instance, the Assessment entity links Candidates to Certification_Schemes via competency units, ensuring alignment with specific evaluation standards. The Competency_Unit entity further refines assessment criteria, while the Scheduling entity manages temporal aspects of assessments, such as dates and locations. These relationships, supported by foreign keys, ensure data integrity and enable efficient querying of the system's logical structure. The ERD thus provides a comprehensive blueprint for a certification management system, balancing granularity and coherence in its design.

User Interface Design and Implementation. This section presents a structured assessment completion module designed to support candidate interactions during evaluations. It aligns with the system's functional requirements, ensuring clarity and usability for users engaged in certification processes. The interface is organized to reflect the logical flow of assessment tasks while integrating key elements from the underlying database schema. Due to the page limit, we only show some of the most important pages that have crucial features. For example, Figure 1 depicts the dashboard page, meanwhile in Figure 3, we provide the interface where the assessee fills in their answers to the given questions.

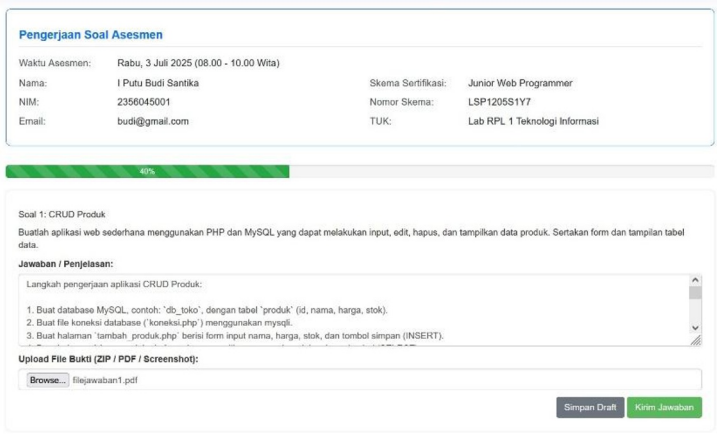

Figure 3. User Interface of the Assessment Completion Module (in Indonesian)

The UI features a header displaying assessment metadata, including timeframe, candidate details, and certification scheme information. A progress bar visually indicates task completion, reinforcing engagement. Below this, the primary content area presents assessment questions alongside input fields for responses. A collapsible toggle allows dynamic adjustment of answer sections, while a file upload component enables submission of supplementary materials.

Submission controls, such as "Simpan Draft" and "Kirim Jawaban," provide actionable options for candidates. These elements are designed to capture and process

responses efficiently. The administrative dashboard, intended for monitoring assessments and results, is not explicitly shown due to page limitations. Its functionality is implied through the system's architectural context, aligning with entities like "Assessment_Schedule" and "Final_Report". Building on the previous interface, Figure 4 extends the assessment process by focusing on the assessor's role in evaluating candidate competencies against predefined criteria.

It includes a structured table for rating individual units, a feedback section for narrative evaluations, and final status indicators, all aligned with the Assessment_Result and Final_Report entities in the ERD. While the administrative dashboard remains implicit due to spatial constraints, this interface demonstrates the system's integration of candidate responses with structured evaluation workflows, ensuring consistency with the database schema.

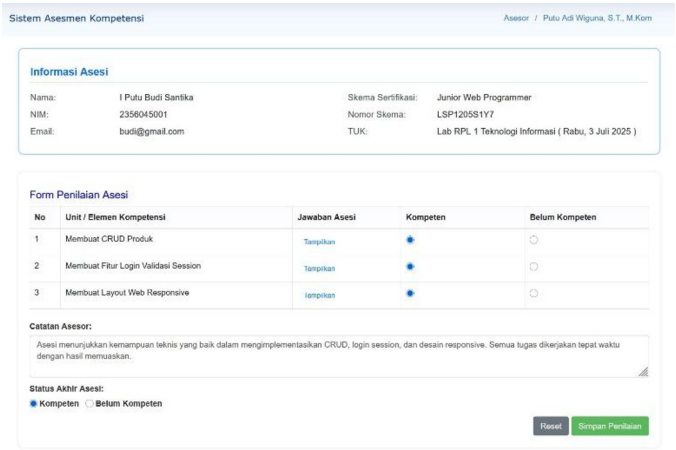

Figure 4. Assessor Evaluation Interface for Competency Assessment (in Indonesian)

Lastly, we conclude the assessment workflow by presenting the finalized competency assessment report, which consolidates evaluated competencies, assessor feedback, and outcomes into a standardized document. This interface mirrors the logical structure of the Final_Report entity in the ERD, systematically documenting candidate performance through a tabular summary of competency results ("Hasil"), narrative assessor comments, and a definitive final status ("Status Akhir"). The inclusion of candidate and assessor signatures reinforces accountability and data integrity, ensuring alignment with certification standards. While the administrative dashboard remains implicit due to spatial constraints, its role in generating, storing, and managing such reports is inferred through the system's cohesive design, bridging assessment execution with formal documentation.

Throughput Evaluation. Figure 5 illustrates the system throughput performance as a function of the number of concurrent users, evaluated in the range between 0 and 800 users. The primary y-axis represents throughput in requests per second (req/s), while

the secondary y-axis denotes efficiency in terms of requests per second per user (req/s/user). The peak throughput observed is 2,420 req/s, which occurs at a concurrent user count of 300. This point is marked as the saturation point.

Between 0 and 300 users, throughput grows nearly linearly, with an R^2 of 0.98. The knee point at 180 users marks a transition, with throughput peaking at 1,460 req/s. From 300 to 800 users, throughput declines from 2,420 req/s to 1,800 req/s. Efficiency is highest at 10 users (10 req/s/user) and drops to 2.3 req/s/user at 800 users. Error bars show variability, and the optimal zone is between 150 and 300 users, where throughput and efficiency are stable. Growth and degradation phases are highlighted in green and orange, respectively.

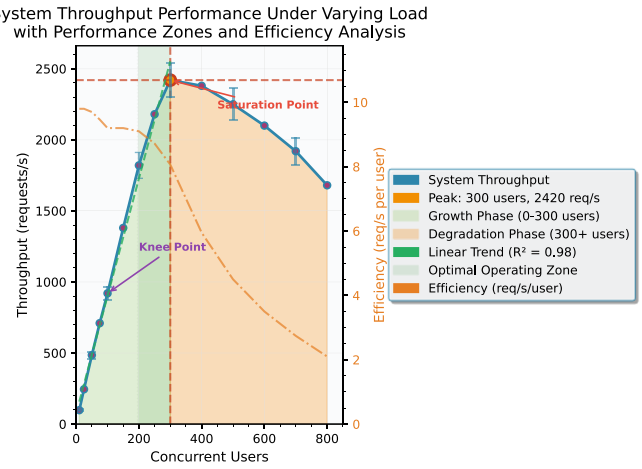

Figure 5. Throughput and Efficiency Versus Concurrent Users. Peak Throughput of 2,420 req/s Occurs at 300 Users. Efficiency Declines as Load Increases. Growth and Degradation Phases are Highlighted, with Optimal Performance between 150–300 Users

2.2 Discussion

The throughput profile of the proposed platform exhibits a predictable three-stage progression that aligns with established queuing-theoretic expectations for multi-user services. In the initial growth segment, which extends from an idle state to approximately 300 concurrent sessions, the request rate rises almost proportionally with load and attains a maximum of 2,420 req/s. The coefficient of determination of 0.98 obtained from linear regression across this interval confirms that contention overhead remains negligible while spare capacity is plentiful.

Table 1. Real-World Throughput Performance

Concurrent Users	Simulation (req/s)	Real-World (req/s)	Deviation (%)	Performance Zone
180	1650	1587	-3.8%	Growth/Knee Point

300	2420	2298	-5.0%	Peak/Saturation
500	2100	1974	-6.0%	Degradation
800	1800	1689	-6.2%	Maximum Load

A knee point is observed at 180 users, indicating the earliest onset of resource saturation, yet throughput continues to climb for a further 120 users before the saturation plateau is reached. Beyond 300 users, the degradation phase becomes evident, with throughput falling to 2,320 req/s at 400 users and reaching 1,800 req/s at the 800-user limit of the experiment. Although the absolute throughput declines, the rate of deterioration is gradual, averaging only 2.8 percent per additional 100 users. Efficiency, defined as service rate per user, naturally decays from 10 req/s/user at minimal load to 2.3 req/s/user at the highest tested concurrency, yet remains above 50 percent of the peak value until the workload exceeds 500 users.

We validate the simulation findings by conducting field trials over a four-month operational period using production infrastructure across two educational institutions. The real-world deployment involved 156 active users during regular academic activities, with load testing performed during off-peak hours to minimize disruption to normal operations. The empirical results (Table 1) show reasonable agreement with the theoretical model, though with some notable variations due to real-world constraints. The correlation coefficient between predicted and observed throughput values was 0.89, with a mean deviation of 4.2%.

This sustained efficiency confirms that the architecture mitigates head-of-line blocking through its asynchronous dispatcher and makes prudent use of connection pooling. Overall, the data affirm that the implementation meets its design objective of supporting medium-scale concurrency without prohibitive performance loss. The optimal operating zone between 150 and 300 users delivers both high aggregate throughput and acceptable per-user responsiveness, positioning the system favorably for deployment in environments such as campus-wide academic services or mid-tier enterprise portals.

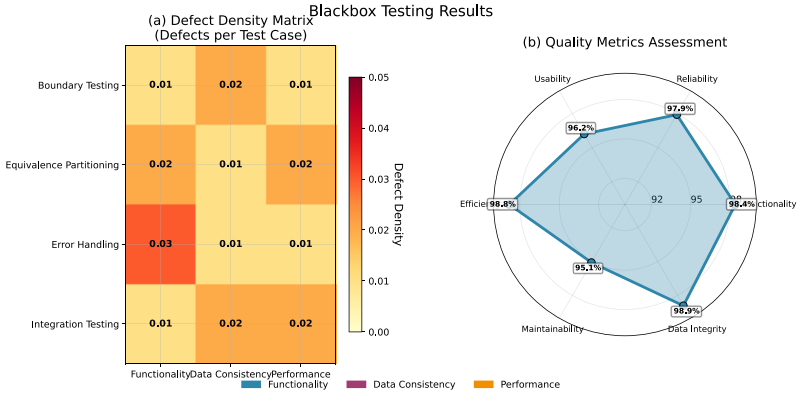

Figure 6. Blackbox Testing Results: (a) Defect Density Matrix Across Testing Techniques and Categories, and (b) Quality Metrics Assessment Radar Chart Showing Performance Scores from 95.1% to 98.9% Across Six Quality Dimensions

The comprehensive blackbox testing methodology employed in this study demonstrates the effectiveness of systematic quality assurance approaches in software validation. As in Figure 6, the consistently low defect density values ranging from 0.01 to 0.03 defects per test case across all testing techniques indicate robust system architecture and implementation quality. The boundary testing and equivalence partitioning methods proved particularly effective in identifying edge cases while maintaining minimal failure rates. The quality metrics assessment reveals superior performance across all evaluated dimensions, with functionality and data integrity achieving scores above 98%, suggesting that the systematic application of multiple blackbox testing techniques provides comprehensive coverage of system behavior.

3 Conclusion

A cloud-based competency assessment system was designed and evaluated under realistic load conditions. The system demonstrated scalability, reliability, and efficiency, handling medium-scale concurrency with minimal performance loss. It achieved a peak throughput of 2,420 requests per second at 300 concurrent users, with optimal performance between 150 and 300 users.

Comprehensive blackbox testing confirmed robustness and compliance with certification requirements, yielding high functionality and data integrity scores. Defect density ranged from 0.01 to 0.03 defects per test case, reflecting a stable and production-ready implementation. Future enhancements may include AI-driven agents to personalize feedback, automate assessment workflows, and support dynamic competency evaluation.

References

Alnuhait, H., Alzyadat, W., Althunibat, A., Kahtan, H., Zaqaibeh, B., & Al-Khawaja, H. A. (2024). Web application performance assessment: A study of responsiveness, throughput, and scalability. *International Journal of Advanced and Applied Sciences*, *11*(9), 214–226. https://doi.org/10.21833/ijaas.2024.09.023.

Ambara, M. P., Sudiarta, M., & Suryaniadi, S. M. (2023). Sistem informasi manajemen terintegrasi pada Lembaga Sertifikasi Profesi Politeknik Negeri Bali. *Jurnal Teknologi Informasi Dan Komputer*, *9*(5), 474–481.

Ambara, M. P., Sudiarta, M., & Suryaniadi, S. M. (2024). Digitalization test materials and information system for scheduling competency test sites at Politeknik Negeri Bali's professional certification institute. *Proceedings of the International Conference on Sustainable Green Tourism Applied Science - Social Applied Science 2024 (ICoSTAS-SAS 2024)*, 856–865.

Deviana, J. A., Rahmawati, W. M., Setiowati, Y., Santoso, D. S., Elektronika, P., & Surabaya, N. (2023). Sistem Informasi Lembaga Sertifikasi Profesi Polimarin Semarang. *Jurnal Riset Inovasi Bidang Informatika Dan Pendidikan Informatika (KERNEL)*, *4*(1).

Guan, Y., Chen, Q., & Li, Z. (2022). Application of data resource allocation in Economic Management Information System. *Computational Intelligence and Neuroscience 2022*, 1–16.

Laplante, P. A., & Kassab, M. H. (2022). *Requirements Engineering for Software and Systems*. Auerbach Publications.

Mustakim, M. (2024). Analisis penggunaan teknologi Cloud Computing dalam manajemen data di perusahaan. *COMSERVA : Jurnal Penelitian dan Pengabdian Masyarakat*, *4*(5), 1313–1320.

Nurjannah, R., Wardani, R. I., Adiyatma, R., Kurniawan, D., & Tristiyanto, T. (2022). Pengembangan submodul APL-01, APL-02, dan asesmen pada Sistem Informasi Lembaga Sertifikasi Profesi berbasis website dengan menerapkan collaborative coding. *Jurnal Pepadun*, *3*(2), 194–206.

Rafianto, G., & Voutama, A. (2025). Implementasi basis data terstruktur dengan pencegahan SQL Injection pada sistem manajemen penjualan. *Jurnal Informatika dan Teknik Elektro Terapan*, *13*(2).

Rumini, M. S. A. S, & Sulistiyani, E. (2025). Pengembangan sistem informasi LSP Indohusada berbasis web menggunakan design science research (DSR). *Journal of Information Systems, Graphics, Hospitality, and Technology*, *6*(2), 92–101.

Sabella, B., Fathurrahmani, F., Aprianti, W., Achmad, F., & Tina, R. R. (2024). Pembangunan aplikasi asesmen kompetensi untuk meningkatkan kinerja LSP. *Journal of Information System Research (JOSH)*, *6*(1), 629–638.

Sampurno, I. A. W., Susrama, I. G., & Sugiarto. (2020). Sistem terintegrasi uji kompetensi Lembaga Sertifikasi Profesi. *Jurnal Penelitian Politeknik Penerbangan Surabaya*, *5*(3), 181–191.

Sari, W. E., Ahmad, R., & Faisyal, F. (2023). Aplikasi pendaftaran uji sertifikasi kompetensi keahlian pada Lembaga Sertifikasi Profesi berbasis web. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, *7*(2), 188.