# Technical Report for SPMB-Online System Development

## 1. Research on SPMB and Development of its Information System

### 1.1 Overview of SPMB (Sistem Penerimaan Murid Baru)

- The SPMB system is primarily used for the admission of new students into schools across Indonesia, regulated by national laws such as Permendikdasmen No. 3/2025.
- SPMB is implemented across various regions, including DKI Jakarta and Surabaya, with significant differences in the technical and policy approaches.

### 1.2 System Requirements and Business Processes

- Key objectives include:
  - Ensuring data integrity, including linking with official government databases like Dukcapil for valid data.
  - Adhering to dynamic regulation changes (e.g., changes in quotas, selection rules).
  - Ensuring flexibility and adaptability in the system's design to accommodate regional differences.

### 1.3 Technical Recommendations for SPMB System Development

- Rules Engine: A highly configurable rules engine is essential, enabling adjustments in selection criteria, quotas, and workflows without requiring code changes.
- Interoperability: Integrating the system with official data sources (e.g., Dukcapil, Dapodik) via APIs ensures data consistency.
- User Experience: Prioritize an intuitive interface and provide accessibility features for a broader audience. Additionally, include support features like AI-powered chatbots.

## 2. Enhancements According to Best Practices

2.1 System Architecture

- Microservices Architecture: Adopting a microservices-based architecture ensures scalability, fault tolerance, and flexibility. Each microservice can handle one business logic unit (e.g., user management, document verification, payments).
- Cloud-Native Infrastructure: Deploying the system on cloud platforms like AWS or Google Cloud ensures scalability with automated resource management during high traffic periods.

2.2 Database Design

- Use of PostgreSQL: A relational database system with support for JSONB to manage semi-structured data. Key tables include applications, users, documents, payments, and admission_cycles.
- Data Integrity and Audit Logs: Key actions should be logged for transparency and auditability, especially actions involving sensitive student data.

2.3 Integration with Third-Party Services

- Payment Gateway Integration: Integrating services like Midtrans for processing payments, ensuring smooth and secure transactions.
- Real-time Notifications: Use Wati API for WhatsApp notifications to communicate updates on status, payment confirmations, and test schedules.

3. Detailed Functional and Non-Functional Requirements

3.1 Core Functionalities

- User Registration: Account creation with email/phone verification, with a focus on data validation against external sources (e.g., Dukcapil).

- Document Upload and Verification: Ensure that documents (KK, Akta, Rapor) are uploaded in the correct format, followed by a validation process by administrators.
- Selection Process: Implement a flexible rules engine to manage various types of selection criteria (e.g., zoning, academic performance, disability status).
- Payment Management: Seamless payment integration for registration and tuition fees.

3.2 Non-Functional Requirements

- Security: Implement multi-factor authentication (MFA), end-to-end encryption, and role-based access control (RBAC) to protect sensitive data.
- Performance: The system must handle high traffic loads during peak registration times, with automated scaling in the cloud environment.
- Compliance with Data Protection Laws: Adherence to Indonesia's UU PDP (Data Protection Law), ensuring that all data handling processes are compliant.

4. Recommendations for Technical Documentation

4.1 Technical Design Documents

- System Architecture: Diagrams of the microservices architecture, API design, and cloud infrastructure.
- Database Schema: Detailed entity-relationship diagrams (ERD) for managing students, documents, and payment data.
- API Specifications: Documentation for third-party integrations, such as payment gateways and notification systems.

4.2 Development Roadmap

- MVP: Focus on core features such as user registration, document upload, and basic payment

functionality.

- Phase 2: Introduce additional features such as rules engine configurations, notifications, and reporting.

- Phase 3: Expand scalability, integrate machine learning for predictive analytics, and develop a mobile app.