

Penelitian ini merupakan pengembangan lanjutan dari sistem pusat data untuk penyimpanan dan pengelolaan data sensor berbasis *Internet of Things* (IoT), dengan fokus pada peningkatan efisiensi pemrosesan data melalui integrasi mekanisme *caching*, modul kueri kompleks, dan pemantauan performa *cluster*. Sistem ini menggunakan *ScyllaDB* sebagai basis data *NoSQL* terdistribusi yang dikombinasikan dengan *DuckDB* sebagai *engine* kueri *SQL in-process*, sehingga data mentah dari *ScyllaDB* dapat dimuat sementara ke *DuckDB* dalam format *JSONL* untuk mendukung analisis fleksibel terhadap data sensor. Pustaka *Moka* diterapkan sebagai *in-memory cache* untuk menyimpan hasil kueri berdasarkan kombinasi ID proyek, ID koleksi, dan perintah *SQL*, dengan dukungan fitur *time-to-live* (TTL) serta mekanisme invalidasi manual saat data diperbarui. Untuk memastikan kesehatan dan kinerja *cluster* secara *real-time*, sistem ini diintegrasikan dengan *Scylla Monitoring Stack*, yang mencakup *Prometheus*, *Grafana*, dan *Alertmanager*, sehingga setiap *alert* dapat dikirimkan ke *administrator* apabila terjadi kondisi abnormal atau penurunan performa. Hasil pengujian menunjukkan bahwa pendekatan ini mampu menurunkan latensi kueri secara signifikan, menjaga stabilitas sistem dalam kondisi beban tinggi, dan menyediakan solusi pusat data yang *scalable*, efisien, serta dapat diakses melalui *REST API*, *MQTT*, dan *WebSocket*.

Kata kunci: pusat data, IoT, *ScyllaDB*, *DuckDB*, *caching*, *SQL*, Rust, *REST API*, *Prometheus*, *Grafana*, *Alertmanager*

This research represents an advanced development of a data center system for storing and managing sensor data based on the Internet of Things (IoT), with a focus on improving data processing efficiency through the integration of caching mechanisms, complex query modules, and cluster performance monitoring. The system employs ScyllaDB as a distributed NoSQL database combined with DuckDB as an in-process SQL query engine, allowing raw data from ScyllaDB to be temporarily loaded into DuckDB in JSON format to support flexible analysis of sensor data. The Moka library is implemented as an in-memory cache to store query results based on the combination of project ID, collection ID, and SQL commands, with support for time-to-live (TTL) and manual invalidation when data is updated. To ensure real-time cluster health and performance monitoring, the system is integrated with the Scylla Monitoring Stack, which includes Prometheus, Grafana, and Alertmanager, so that any alert can be sent to the administrator in case of abnormal conditions or performance degradation. Testing results indicate that this approach significantly reduces query latency, maintains system stability under high load, and provides a scalable and efficient data center solution accessible via REST API, MQTT, and WebSocket.

Keywords: data center, IoT, ScyllaDB, DuckDB, caching, SQL, Rust, REST API, monitoring, alerting