Pengembangan Sistem *Backend* Koperasi Terpusat Berbasis *Rest Api Node.Js* Untuk Pengelolaan Transaksi Dan Percepatan Rekapitulasi Laporan Keuangan

[1st]Yogi Hafidh Maulana
Telkom University
Software Engineering
Purwokerto, Indonesia
hafidhyogi@gmail.com

[2nd]Arif Amrulloh
Telkom University
Software Engineering
Purwokerto, Indonesia
amrulloh@telkomuniversity.ac.id

[3nd]Dwi Mustika Kusumawardani
Telkom University
Sistem Informasi
Purwokerto, Indonesia
dwimustika@telkomuniversity.ac.id

Abstrak — Koperasi Simpan Pinjam PRIMKOPPABRI Kusuma Bangsa masih mengelola data transaksi secara manual menggunakan buku tulis, sehingga menyebabkan inkonsistensi data, kesalahan pencatatan, dan keterlambatan penyusunan laporan keuangan hingga tiga hari. Penelitian ini bertujuan mengembangkan sistem *backend* terpusat berbasis *REST API* untuk menyediakan data transaksi secara *real-time* guna mendukung monitoring keuangan dan percepatan pelaporan. Metode yang digunakan adalah *Agile* dengan kerangka kerja *Scrum* yang dilaksanakan dalam empat *sprint*, serta pengujian sistem melalui *black-box testing* dan *integration testing*. Hasil penelitian menunjukkan bahwa sistem berhasil mengintegrasikan sepuluh modul operasional koperasi ke dalam satu basis data terpusat, dilengkapi dengan mekanisme kontrol akses dan pencatatan aktivitas pengguna. Seluruh 179 *testcase* pada pengujian fungsional dan 153 *testcase* pada pengujian integrasi berhasil dijalankan tanpa kegagalan, yang menunjukkan tingkat keandalan sistem yang tinggi. Implementasi sistem ini mampu meningkatkan efisiensi operasional dengan menyediakan informasi keuangan yang lebih akurat serta mempercepat proses pelaporan dari tiga hari menjadi *real-time*.

Kata kunci — backend, koperasi, Scrum, pengujian sistem, real-time, integrasi

I. PENDAHULUAN

Perkembangan teknologi informasi pada era Revolusi Industri 4.0 mendorong organisasi untuk melakukan transformasi digital guna meningkatkan efisiensi, transparansi, dan akuntabilitas operasional [1], [2], [3]. Dalam konteks koperasi, digitalisasi tidak lagi menjadi pilihan, melainkan kebutuhan untuk menjaga keberlanjutan usaha dan meningkatkan kualitas layanan [4]. Berbagai penelitian telah mengembangkan sistem informasi koperasi berbasis web, khususnya pada aspek pencatatan transaksi dan pelaporan keuangan [5]. Namun, sebagian besar solusi tersebut masih berfokus pada digitalisasi proses tanpa memperhatikan arsitektur *backend* yang terintegrasi, terverifikasi, dan dapat diaudit [6], sehingga permasalahan terkait akurasi, konsistensi, dan keamanan data masih belum terselesaikan secara optimal [7].

Ketiadaan arsitektur *backend* terpusat menyebabkan terjadinya fragmentasi data dan inkonsistensi informasi, terutama pada koperasi yang masih mengandalkan pencatatan manual [12]. Pengelolaan data secara manual rentan terhadap kesalahan input, duplikasi data, serta kekeliruan perhitungan transaksi [8], [9]. Dampaknya, proses pencarian data dan penyusunan laporan keuangan menjadi lambat karena memerlukan rekapitulasi lintas catatan, sehingga menurunkan akurasi laporan dan menghambat pengambilan keputusan secara tepat waktu [10], [11]. Berdasarkan hasil observasi dan wawancara di Koperasi Simpan Pinjam PRIMKOPPABRI Kusuma Bangsa, proses pengelolaan transaksi masih dilakukan secara manual menggunakan buku tulis, sehingga laporan keuangan baru dapat disusun setelah proses rekapitulasi berlapis yang memerlukan waktu hingga ±3 hari untuk lebih dari 200 nasabah.

Permasalahan tersebut diperparah oleh belum adanya mekanisme validasi data otomatis, *Role-Based Access Control (RBAC)*, serta *audit trail* yang berfungsi untuk menjaga integritas dan keamanan data [13], [14]. Selain itu, belum diterapkannya arsitektur modular berbasis *RESTful API* menyebabkan keterbatasan dalam integrasi dan interoperabilitas sistem [15]. Padahal, penelitian terdahulu menunjukkan bahwa penerapan sistem informasi terintegrasi mampu meningkatkan transparansi, akurasi laporan, serta efisiensi operasional melalui percepatan aliran informasi dan pengurangan redundansi data [17], [18].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan sistem *backend* koperasi digital berbasis arsitektur *RESTful API* menggunakan *Node.js* dan *framework NestJS*. Sistem dirancang sebagai *single source of truth* yang mampu mengelola dan menyajikan data transaksi secara *real-time*, serta dilengkapi dengan mekanisme autentikasi *JWT*, kontrol akses berbasis peran, dan *audit trail* untuk menjamin keamanan dan integritas data [21]. Pengembangan sistem dilakukan menggunakan metode *Agile* dengan kerangka kerja *Scrum* yang memungkinkan

pengembangan secara iteratif dan inkremental [22], [23], dengan pembagian modul yang terstruktur agar sistem lebih modular dan mudah dikembangkan [24]. Evaluasi sistem dilakukan melalui pengujian *black-box testing* untuk memastikan kesesuaian fungsionalitas *API* serta *integration testing* untuk memverifikasi keterpaduan antar modul dalam sistem [25], [26], [27], [28]. Dengan pendekatan ini, diharapkan sistem mampu meningkatkan efisiensi operasional koperasi, mempercepat proses pelaporan, serta menyediakan informasi keuangan yang akurat dan tepat waktu.

II. KAJIAN TEORI

A. Koperasi dan Sistem Informasi Koperasi

Koperasi merupakan badan usaha berbasis prinsip ekonomi kerakyatan yang bertujuan meningkatkan kesejahteraan anggota melalui aktivitas penghimpunan dan penyaluran dana [38], [39]. Dalam koperasi simpan pinjam, proses bisnis utama meliputi pengelolaan simpanan, pinjaman, serta angsuran yang seluruhnya bermuara pada penyusunan laporan keuangan. Pengelolaan secara manual rentan terhadap kesalahan pencatatan, keterlambatan pelaporan, serta inkonsistensi data [40]. Oleh karena itu, sistem informasi koperasi dibutuhkan untuk mengintegrasikan proses bisnis dan menghasilkan informasi yang akurat, *real-time*, dan terstruktur guna mendukung pengambilan keputusan.

B. *Backend Development* dan *RESTful API*

Backend merupakan komponen sisi server yang bertanggung jawab dalam pengolahan logika bisnis, manajemen basis data, serta penyediaan layanan melalui *Application Programming Interface (API)* [21], [44]. *RESTful API* merupakan gaya arsitektur yang memanfaatkan protokol HTTP dengan metode standar seperti GET, POST, PUT, dan DELETE untuk pertukaran data secara stateless [45]. Penggunaan format *JSON* menjadikan komunikasi data lebih ringan dan efisien, sehingga *REST API* menjadi standar dalam pengembangan sistem modern yang membutuhkan interoperabilitas tinggi.

C. Arsitektur Sistem *Backend*

Pendekatan arsitektur yang digunakan dalam pengembangan *backend* modern salah satunya adalah *Modular Monolith*, yaitu pengembangan sistem dalam satu aplikasi tetapi terstruktur ke dalam modul-modul yang independen [37], [46]. Pendekatan ini memberikan keseimbangan antara kesederhanaan deployment dan *maintainability* sistem. Selain itu, *Clean Architectur*e diterapkan untuk memisahkan logika bisnis dari detail implementasi, sehingga sistem menjadi lebih fleksibel, mudah diuji, dan scalable [44].

D. Teknologi *Backend*: *Node.js* dan *NestJS*

Node.js merupakan runtime *JavaScript* berbasis *event-driven dan non-blocking I/O* yang mampu menangani banyak permintaan secara efisien [45]. Sementara itu, *NestJS* merupakan *framework backend* berbasis *Node.js* yang mendukung arsitektur modular, *dependency injection*, serta penggunaan *TypeScript* untuk meningkatkan struktur dan *maintainability* kode [21], [37]. Kombinasi teknologi ini memungkinkan pengembangan sistem *backend* yang scalable, terstruktur, dan efisien.

E. Keamanan Sistem *Backend*

Keamanan sistem *backend* menjadi aspek penting dalam menjaga integritas data. *JSON Web Token (JWT)* digunakan sebagai mekanisme autentikasi berbasis *stateless* untuk memverifikasi identitas pengguna [47]. Selain itu, *Role-Based Access Control* (RBAC) digunakan untuk membatasi akses sistem berdasarkan peran pengguna [48]. Implementasi validasi input, *middleware* keamanan, serta *audit trail* juga diperlukan untuk mencegah manipulasi data dan mencatat aktivitas pengguna dalam sistem.

F. Metode Pengembangan *Agile* dan *Scrum*

Agile merupakan pendekatan pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi, dan iterasi berkelanjutan [49]. Salah satu *framework* yang umum digunakan adalah *Scrum*, yang membagi proses pengembangan ke dalam siklus *sprint* untuk menghasilkan *increment* sistem secara bertahap [50]. *Scrum* memungkinkan tim untuk beradaptasi terhadap perubahan kebutuhan serta meningkatkan kualitas sistem melalui evaluasi berkelanjutan.

G. *Database Management System (DBMS)*

Database merupakan kumpulan data terstruktur yang dikelola oleh *Database Management System (DBMS)* untuk mendukung penyimpanan dan pengolahan data secara efisien [51]. *Relational Database Management System (RDBMS)* seperti *PostgreSQL* banyak digunakan karena mendukung integritas data melalui relasi tabel serta prinsip *ACID (Atomicity, Consistency, Isolation, Durability)* [52]. *PostgreSQL* memiliki keunggulan dalam menangani transaksi kompleks dan konkurensi tinggi, sehingga sesuai untuk sistem keuangan seperti koperasi [53].

H. Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk memastikan sistem berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan [25]. *Black-box testing* digunakan untuk menguji fungsionalitas sistem berdasarkan *input* dan *output* tanpa melihat kode program [25], [34]. Sementara itu, *integration testing* digunakan untuk memastikan bahwa modul-modul dalam sistem dapat terintegrasi dengan baik dan berfungsi secara keseluruhan [27], [59]. Kombinasi kedua metode ini penting untuk menjamin kualitas dan keandalan sistem *backend*.

Berbagai penelitian sebelumnya telah mengembangkan sistem informasi koperasi berbasis web untuk menggantikan proses manual [5], [29]. Selain itu, penelitian lain menunjukkan bahwa penerapan *RESTful API* mampu meningkatkan efisiensi komunikasi data dan performa sistem [30], serta penggunaan *Node.js* memberikan performa yang responsif dalam pengembangan layanan API [32]. Di sisi lain, penelitian terkait *Scrum* menunjukkan bahwa pendekatan iteratif mampu meningkatkan fleksibilitas dan produktivitas pengembangan [33]. Namun, sebagian besar penelitian masih berfokus pada pengembangan sistem secara umum (fullstack) tanpa menitikberatkan pada arsitektur *backend* yang terintegrasi, aman, dan terverifikasi. Oleh

karena itu, penelitian ini mengisi celah tersebut dengan mengembangkan sistem *backend* koperasi berbasis *REST API* yang modular, aman, dan teruji.

III. METODE

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan metode *Agile* melalui kerangka kerja *Scrum* untuk mengembangkan sistem *backend* koperasi digital. Proses penelitian dimulai dari identifikasi masalah pada pengelolaan data transaksi di Koperasi Simpan Pinjam PRIMKOPPABRI Kusuma Bangsa yang masih dilakukan secara manual, kemudian dilanjutkan dengan studi literatur untuk memperoleh landasan teoritis terkait sistem informasi koperasi, arsitektur *backend*, *RESTful API*, serta keamanan sistem.

Sumber data dalam penelitian ini terdiri dari data primer dan data sekunder. Data primer diperoleh melalui wawancara semi-terstruktur dengan pimpinan dan pegawai koperasi, serta observasi non-partisipan terhadap proses operasional yang berjalan. Wawancara digunakan untuk menggali kebutuhan sistem dan aturan bisnis, sedangkan observasi dilakukan untuk memahami alur pencatatan transaksi dan permasalahan yang terjadi. Data sekunder diperoleh dari jurnal ilmiah, buku, serta dokumentasi teknologi yang relevan untuk mendukung perancangan sistem.

Berdasarkan data yang diperoleh, dilakukan analisis kebutuhan sistem yang mencakup kebutuhan fungsional dan non-fungsional. Selanjutnya, dilakukan perancangan sistem *backend* yang meliputi pemodelan sistem menggunakan *Unified Modeling Language (UML)*, perancangan basis data menggunakan *Entity Relationship Diagram (ERD)*, serta perancangan arsitektur sistem berbasis *RESTful API* dengan pendekatan modular. Sistem dikembangkan menggunakan *Node.js* dengan *framework NestJS*, database *PostgreSQL*, serta *ORM Prisma* untuk pengelolaan data.

Proses pengembangan sistem dilakukan secara iteratif dalam empat *sprint* menggunakan *Scrum*. Setiap *sprint* meliputi tahap perencanaan (*Sprint* Planning), pengembangan, pengujian, serta evaluasi melalui *Sprint Review* dan *Sprint Retrospective*. Pembagian *sprint* dilakukan berdasarkan modul sistem, yaitu modul autentikasi dan otorisasi, modul nasabah dan transaksi, modul simpanan dan pinjaman, serta modul laporan dan fitur pendukung seperti dashboard dan audit trail.

Pengujian sistem dilakukan menggunakan dua metode utama, yaitu *black-box testing* dan *integration testing*. *Black-box testing* digunakan untuk menguji fungsionalitas setiap endpoint API berdasarkan kesesuaian input dan output tanpa melihat struktur kode internal [25]. Tingkat keberhasilan pengujian dihitung menggunakan persamaan berikut:

$$\text{Presentase Validasi} = \left(\frac{jumlah\ testcase\ valid}{total\ testcase}\right) \times 100\%$$

IV. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sistem *backend* koperasi digital yang dikembangkan menggunakan pendekatan *Agile Scrum* dalam empat *sprint*. Setiap *sprint* menghasilkan modul sistem yang terintegrasi secara bertahap, mencakup autentikasi dan otorisasi, pengelolaan data nasabah dan transaksi, pengelolaan simpanan dan pinjaman, serta penyajian laporan keuangan dan monitoring sistem.

A. Hasil Pengembangan Sistem

Sistem *backend* koperasi digital berhasil dikembangkan menggunakan arsitektur modular berbasis *RESTful API* yang terdiri dari beberapa lapisan utama, yaitu *controller*, *service*, dan *database*. Arsitektur ini memungkinkan pemisahan tanggung jawab antar komponen sehingga sistem menjadi lebih terstruktur dan mudah dikembangkan.

Hasil pengembangan menunjukkan bahwa seluruh modul sistem *backend* berhasil diimplementasikan sesuai dengan kebutuhan operasional koperasi. Modul autentikasi dan otorisasi mampu mengelola akses pengguna menggunakan mekanisme *JSON Web Token (JWT)* dan *Role-Based Access Control (RBAC),* sehingga setiap pengguna hanya dapat mengakses fitur sesuai hak aksesnya.

Pada modul operasional, sistem mampu mengelola data nasabah, transaksi, simpanan, dan pinjaman secara terintegrasi. Setiap transaksi yang dilakukan akan langsung tercatat dalam basis data dan dapat digunakan sebagai dasar penyusunan laporan keuangan. Hal ini menunjukkan bahwa sistem telah menerapkan konsep *single source of truth* dalam pengelolaan data.

Selain itu, modul laporan keuangan mampu menghasilkan rekapitulasi data secara otomatis dan *real-time*, yang sebelumnya dilakukan secara manual dan memerlukan waktu 2–3 hari. Dengan sistem yang dikembangkan, proses tersebut dapat dilakukan secara lebih cepat dan akurat.

Modul pendukung seperti dashboard dan *audit trail* juga berhasil diimplementasikan untuk memberikan informasi ringkas kondisi operasional serta pencatatan aktivitas perubahan data. Hal ini meningkatkan transparansi dan kemudahan monitoring sistem.

B. Hasil Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *black-box testing* dan *integration testing* untuk memastikan kesesuaian fungsi serta integrasi antar modul.

1. Blackbox Testing

Berdasarkan hasil pengujian, seluruh endpoint API telah diuji menggunakan berbagai skenario *input*. Total pengujian mencapai 179 *test case*, dan seluruhnya berhasil dijalankan tanpa kegagalan. Ringkasan hasil *blackbox testing* dapat dilihat pada tabel 1 dibawah ini.

TABEL 1
(Ringkasan Hasil *Blackbox Testing*)

Parameter	Nilai
Total Test Case	179
Berhasil	179
Gagal	0
Persentase Keberhasilan	100%

Hasil ini menunjukkan bahwa sistem mampu memvalidasi input dengan baik, menghasilkan *output* sesuai spesifikasi, dan menangani *error* sesuai aturan bisnis.

2. *Integration testing*

Integration testing dilakukan untuk memastikan keterhubungan antar modul sistem. Total pengujian mencapai 153 *test case*, dengan seluruhnya berhasil dijalankan tanpa kegagalan. Ringkasan hasil *Integration testing* dapat dilihat pada tabel 2 dibawah ini.

TABEL 2
(Ringkasan Hasil *Integration Testing*)

Parameter	Nilai
Total Test Case	153
Berhasil	153
Gagal	0
Persentase Keberhasilan	100%

Hasil ini menunjukkan bahwa modul sistem terintegrasi dengan baik, tidak terjadi konflik data antar modul, dan konsistensi data terjaga.

C. Pembahasan

Hasil penelitian menunjukkan bahwa penerapan arsitektur *backend* berbasis *RESTful API* dengan pendekatan modular mampu mengatasi permasalahan utama pada sistem manual koperasi, yaitu keterlambatan pengolahan data, ketidakakuratan pencatatan, dan kurangnya integrasi antar proses. Sebelum sistem dikembangkan, proses pencatatan dilakukan secara manual menggunakan buku, yang rentan terhadap kesalahan pencatatan dan membutuhkan waktu lama dalam penyusunan laporan. Setelah implementasi sistem, seluruh data tersimpan dalam basis data terpusat dan dapat diakses secara *real-time*, sehingga meningkatkan efisiensi operasional.

Dari sisi kualitas sistem, hasil pengujian menunjukkan tingkat keberhasilan sebesar 100% pada seluruh skenario pengujian. Hal ini mengindikasikan bahwa sistem memiliki tingkat keandalan yang tinggi, baik dari sisi validasi input maupun integrasi antar modul.

Pendekatan *Agile Scrum* juga terbukti efektif dalam pengembangan sistem, karena memungkinkan pengembangan dilakukan secara bertahap dan adaptif terhadap kompleksitas sistem yang meningkat. Setiap *sprint* menghasilkan peningkatan kualitas sistem secara berkelanjutan hingga seluruh modul dapat terintegrasi dengan baik.

D. Generalisasi dan Implikasi

Berdasarkan hasil penelitian, dapat digeneralisasikan bahwa:

1. Penggunaan arsitektur *backend* terpusat mampu meningkatkan konsistensi dan integritas data pada sistem koperasi.
2. Pendekatan modular memudahkan pengembangan dan integrasi sistem yang kompleks.
3. Implementasi *RBAC* dan *audit trail* meningkatkan keamanan serta transparansi sistem.
4. Digitalisasi proses bisnis koperasi dapat meningkatkan efisiensi operasional dan kecepatan penyajian informasi.

E. Pengembangan Selanjutnya

Penelitian ini masih memiliki beberapa keterbatasan, sehingga dapat dikembangkan lebih lanjut, antara lain:

1. Pengembangan antarmuka frontend untuk mendukung sistem *backend* secara menyeluruh
2. Implementasi *notifikasi real-time* (misalnya menggunakan WebSocket)
3. Integrasi dengan sistem pembayaran digital
4. Pengujian performa sistem (*load testing dan stress testing*)
5. Pengembangan analitik data untuk mendukung pengambilan keputusan

V. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem *backend* koperasi digital berbasis arsitektur *RESTful API* menggunakan pendekatan *Agile Scrum* yang mampu menjawab permasalahan operasional pada Koperasi Simpan Pinjam PRIMKOPPABRI Kusuma Bangsa. Sistem yang dibangun mampu menggantikan proses manual berlapis dalam monitoring dan pengelolaan data transaksi menjadi proses terintegrasi dalam satu sistem terpusat. Implementasi sistem memungkinkan penyajian informasi keuangan secara *real-time* dalam waktu kurang dari satu menit, sehingga pimpinan dapat melakukan monitoring dan pengambilan keputusan secara lebih cepat, tepat, dan berbasis data.

Selain itu, sistem *backend* yang dikembangkan juga berhasil meningkatkan efisiensi dalam proses penyusunan dan rekapitulasi laporan keuangan yang sebelumnya membutuhkan waktu hingga ±3 hari menjadi otomatis dan *real-time*. Hasil pengujian menunjukkan tingkat keberhasilan 100% pada seluruh skenario *black-box testing* dan *integration testing*, yang mengindikasikan bahwa sistem memiliki tingkat keandalan yang tinggi baik dari sisi fungsional maupun integrasi antar modul. Dengan demikian, penerapan sistem ini tidak hanya meningkatkan akurasi dan konsistensi data, tetapi juga mendukung efektivitas operasional koperasi secara keseluruhan.

REFERENSI

[1] O. Koperasi Pamekasan *et al.*, “Penerapan teknologi informasi dalam manajemen,” *Jurnal Manajemen Akuntansi*, vol. 2, no. 6, pp. 3031–5220, 2024.

[2] N. M. Sarminiati, P. Anugrah, C. Dewi, dan D. N. Estiyanti, “Analisis tata kelola sistem informasi koperasi

simpan pinjam berbasis *framework* COBIT 5," vol. 11, no. 3, Dec. 2022.

[3] M. F. A. Zakka dan A. Rizaldi, "Eksistensi koperasi yang menjadi tantangan di era revolusi industri 4.0," *JPSDa*, vol. 2, no. 2, pp. 138–148, Jul. 2022.

[4] J. Sewaka Bhakti *et al.*, "Pemanfaatan teknologi digital dalam upaya memperkuat ekonomi digital pada koperasi simpan pinjam," vol. 10, no. 1, Apr. 2024.

[5] D. Kustiawan *et al.*, "Rancang bangun sistem informasi akuntansi pengelolaan koperasi menggunakan metode extreme programming," *Jurnal Teknologi dan Informasi*, vol. 12, no. 1, Apr. 2022.

[6] B. Rudianto dan Y. E. Achyani, "Rancang bangun sistem informasi simpan pinjam pada koperasi berbasis web," *JISAMAR*, vol. 6, no. 1, Feb. 2022.

[7] E. W. Budiningrum dan A. Subiyantoro, "Sistem informasi akuntansi koperasi untuk meningkatkan daya saing," vol. 2, no. 3, Mar. 2023.

[8] K. R. Putratama *et al.*, "Sistem inventory koperasi berbasis website," *JIKTI*, vol. 2, no. 2, 2025.

[9] I. Finesti, R. L. P. Harjanti, dan F. Apriani, "Rancang bangun sistem koperasi simpan pinjam berbasis web prototyping," 2024.

[10] F. Indrawan, E. Darmanto, dan F. Nugraha, "Rancang bangun sistem informasi koperasi simpan pinjam pegawai," *JEKIN*, vol. 5, no. 1, Feb. 2025.

[11] E. Endaryati *et al.*, "Sistem manajemen perhitungan bunga pinjaman," *Jurnal Ilmiah Sains Teknologi dan Informasi*, vol. 2, no. 3, Jun. 2024.

[12] C. Pratiwi dan A. T. Hidayat, "Pengembangan aplikasi manajemen keuangan koperasi menggunakan metode FAST," 2024.

[13] S. Selviani *et al.*, "Pengembangan model integrasi basis data," *RIGGS*, vol. 4, no. 2, Jul. 2025.

[14] M. A. M. Al'Amin *et al.*, "Implementasi sistem informasi koperasi menggunakan CodeIgniter," *RIGGS*, vol. 4, no. 2, May 2025.

[15] I. Niswati dan D. Marlina, "Sistem informasi berbasis *RESTful API*," Aug. 2025.

[16] Z. Kahfi *et al.*, "Edukasi sistem informasi akuntansi koperasi," *JTCSA*, vol. 5, no. 2, Jul. 2025.

[17] B. Wirawan *et al.*, "Pengaruh sistem informasi manajemen terhadap transparansi koperasi," vol. 3, no. 1, May 2025.

[18] L. Siregar dan R. Firdaus, "Pengaruh sistem informasi akuntansi terhadap efisiensi operasional," vol. 1, no. 9, Nov. 2024.

[19] A. A. Hakim *et al.*, "Implementasi *RESTful API* web," 2025.

[20] H. F. Herdiyatmoko dan Y. D. Pratama, "REST API pada sistem transaksi," 2024.

[21] I. R. D. Muhammad dan I. V. Paputungan, "Development of *backend* server based on REST API architecture," *SNATI*, vol. 3, no. 2, pp. 79–87, Jan. 2024.

[22] D. Giawa dan S. Abdy, "Sistem informasi berbasis *Scrum*," *Jurnal Informatika Press*, vol. 1, no. 2, 2024.

[23] R. Noveandini *et al.*, "Penerapan *Scrum* pada sistem informasi," *SeNTIK*, vol. 7, no. 1, 2023.

[24] F. Suarezsaga *et al.*, "Pengembangan sistem menggunakan *Scrum*," Nov. 2022.

[25] M. N. Ichsanudin *et al.*, "Pengujian perangkat lunak dengan metode Black Box Testing," vol. 1, no. 2, 2022.

[26] M. T. Abdillah *et al.*, "Implementasi Black Box Testing pada website," vol. 8, no. 1, 2023.

[27] A. Valerian *et al.*, "Implementasi *integration testing* pada aplikasi web," 2025.

[28] A. U. Afidah *et al.*, "Studi komparasi teknik pengujian end-to-end," vol. 5, no. 2, 2024.

[29] S. I. Simpan *et al.*, "Implementasi sistem informasi koperasi berbasis web," Dec. 2024.

[30] S. Tahseen, "A case study on simplifying mobile integration using JAX-RS," IEEE, Aug. 2025.

[31] J. E. Firdaus dan Y. Prayudi, "Pengembangan *backend* menggunakan *Scrum*," *EDUSAINTEK*, vol. 11, no. 2, Sep. 2023.

[32] B. Kasemsontitum dan T. Tanadechopon, "Performance evaluation of programming languages," IEEE, Jan. 2024.

[33] A. C. Sassa *et al.*, "*Scrum*: a systematic literature review," 2023.

[34] R. G. Guntara dan V. Azkarin, "Implementasi REST API dengan Black Box Testing," *Jurnal Minfo Polgan*, vol. 12, no. 1, Jul. 2023.

[35] O. Zainuddin dan B. Yusof, "Ensuring data integrity in financial systems," 2025.

[36] M. I. P. Nasution, "Manajemen data berbasis database," Universitas Islam Negeri Sumatera Utara, 2025.

[37] H. Fadli dan A. Sukendar, "Pengembangan aplikasi menggunakan Next.js dan *NestJS*," Jul. 2025.

[38] D. M. Situmorang, "Analisis kinerja keuangan koperasi," *Journal Akun Nabelo*, vol. 5, no. 2, Jan. 2023.

[39] S. B. Utomo *et al.*, "Pelatihan digitalisasi koperasi," *Jurnal Pengabdian Masyarakat*, vol. 2, no. 1, Feb. 2022.

[40] B. P. Cahyono dan N. Z. Al-Asfahani, "Implementasi digitalisasi koperasi," 2023.

[41] A. Rivaldi *et al.*, "Metode pengumpulan data melalui wawancara," 2023.

[42] R. Raudah *et al.*, "Analisis dampak keterbatasan sumber daya," *MARAS*, vol. 3, no. 2, May 2025.

[43] Y. M. W. dan T. O. N. K. Putrianto, "Analisis kebutuhan pengguna," vol. 7, no. 1, Mar. 2024.

[44] R. Annisa *et al.*, "Implementasi Golang Clean Architecture," *JITET*, vol. 12, no. 2, Apr. 2024.

[45] T. Kemendikbud *et al.*, "Analisis performa *framework NestJS*," *J-INTECH*, vol. 12, no. 1, Jul. 2024.

[46] M. Ilham *et al.*, "Arsitektur sistem terdistribusi," May 2025.

[47] I. Shikhverdiyev *et al.*, "Secure authentication in e-government 2.0," *Int. Sci. J.*, vol. 3, no. 6, Dec. 2024.

[48] A. M. Harahap, "Implementasi *Role-Based Access Control*," 2024.

[49] Project Management Institute, *Agile Practice Guide*. Newtown Square, PA: PMI, 2017.

[50] J. Sutherland, *Scrum: The Art of Doing Twice the Work in Half the Time*. New York: Crown Business, 2014.

[51] K. Y. Santika *et al.*, "Systematic literature review untuk DBMS," Aug. 2023.

[52] S. Akinola, "Trends in opensource RDBMS," *J. Res. Sci. Eng.*, vol. 6, no. 7, Jul. 2024.

[53] S. V. Salunke dan A. Ouda, "Performance benchmark PostgreSQL dan MySQL," MDPI, Oct. 2024.

[54] S. M. Pulungan *et al.*, "Analisis ERD dalam perancangan database," 2023.

[55] D. F. M. Dina, "Normalisasi database," 2022.

[56] S. W. Ramdany *et al.*, “Penerapan UML Class Diagram,” Jun. 2024.

[57] M. S. Harlina *et al.*, “Pemodelan sistem menggunakan UML,” *JTSIB*, vol. 7, no. 3, Jul. 2025.

[58] B. Hardika *et al.*, “Pengujian Black Box Testing menggunakan equivalence partitioning,” Dec. 2024.

[59] N. N. I. A. R. P. P. Y. P. W. S. K. M. A. R. Hidayat, “Pengujian perangkat lunak lengkap,” vol. 9, no. 4, Aug. 2025.