Teknik: Jurnal Ilmu Teknik dan Informatika
Volume 6 Nomor 2 Oktober 2026
e-ISSN: 2798-2513; p-ISSN: 2808-8751, Hal *98-119*
DOI: https://doi.org/10.51903/teknik.v6i2.1274
Available online at: https://journal.stiestekom.ac.id/index.php/TEKNIK

SmartQBank: Sistem Manajemen Antrean Perbankan Berbasis *Finite State Machine* dengan Sinkronisasi *Realtime* dan Pelacakan QR

Micael Zecsen Saragih[1] Zulfahmi Indra[2] Benedict Sandi Pangestu Rosa[3] Khairida Octavia Ramadhani[4]
[1-4]Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan

Korespondensi penulis: micaelzecsen.424325056@mhs.unimed.ac.id

Abstract.*Conventional banking queue systems still face significant limitations in terms of efficiency, flexibility, and service transparency because customers are required to be physically present in waiting areas and have limited access to real-time queue information. This study aims to design and implement SmartQBank, a digital banking queue management system that integrates Finite State Machine (FSM), cloud computing, real-time synchronization, and Quick Response (QR)-based tracking to improve queue service effectiveness. The research employed an applied software engineering approach consisting of requirements analysis, formal FSM modeling, system design, implementation, and validation testing. The FSM model was utilized as the core engine to control and validate queue lifecycle transitions, ensuring that each queue token follows predefined business rules and prevent ing invalid state changes. The system was developed using Next.js and TypeScript on the frontend, while Supabase PostgreSQL served as the backend database and persistence layer. Real-time synchronization was achieved through WebSocket-based communication provided by Supabase Realtime, enabling instant data propagation across multiple client devices. The implementation results demonstrate that the proposed architecture successfully supports distributed queue monitoring, dynamic QR-based customer tracking, and centralized state management. Validation testing confirmed that all defined FSM transitions operated correctly and consistently rejected invalid transitions, thereby maintaining data integrity and operational reliability. The integration of FSM-based control, cloud infrastructure, and real-time synchronization provides a scalable and reliable solution for modern banking queue management while improving customer convenience, service transparency, and overall operational efficiency.*
Keywords: *cloud computing, Finite State Machine, QR tracking, queue management system, realtime synchronization*

Abstrak. Sistem antrean perbankan konvensional masih menghadapi berbagai keterbatasan dalam aspek efisiensi, fleksibilitas, dan transparansi layanan karena nasabah harus hadir secara fisik di area tunggu serta memiliki akses yang terbatas terhadap informasi antrean secara *realtime*. Penelitian ini bertujuan untuk merancang dan mengimplementasikan *SmartQBank*, yaitu sistem manajemen antrean perbankan digital yang mengintegrasikan *Finite State Machine* (FSM), komputasi awan, sinkronisasi *realtime*, dan pelacakan berbasis Quick Response (QR) untuk meningkatkan efektivitas layanan antrean. Penelitian menggunakan pendekatan rekayasa perangkat lunak terapan yang mencakup tahap analisis kebutuhan, pemodelan formal FSM, perancangan sistem, implementasi, dan pengujian validasi. Model FSM digunakan sebagai mesin inti untuk mengendalikan dan memvalidasi transisi siklus hidup antrean sehingga setiap token antrean mengikuti aturan bisnis yang telah ditetapkan dan terhindar dari perubahan status yang tidak valid. Sistem dikembangkan menggunakan Next.js dan TypeScript pada sisi frontend, sedangkan Supabase PostgreSQL digunakan sebagai basis data dan lapisan persistensi. Sinkronisasi *realtime* diimplementasikan melalui komunikasi berbasis *WebSocket* menggunakan layanan Supabase *Realtime* yang memungkinkan distribusi data secara instan ke berbagai perangkat klien. Hasil implementasi menunjukkan bahwa arsitektur yang diusulkan mampu mendukung pemantauan antrean terdistribusi, pelacakan nasabah berbasis QR dinamis, serta pengelolaan status antrean secara terpusat. Pengujian validasi membuktikan bahwa seluruh transisi FSM berjalan sesuai aturan yang ditetapkan dan secara konsisten menolak transisi yang tidak valid sehingga integritas data dan keandalan operasional tetap terjaga. Integrasi antara kontrol berbasis FSM, infrastruktur komputasi awan, dan sinkronisasi *realtime* menghasilkan solusi manajemen antrean perbankan yang skalabel, andal, serta mampu meningkatkan kenyamanan nasabah, transparansi layanan, dan efisiensi operasional secara keseluruhan.
Kata Kunci: *Finite State Machine*, komputasi awan, manajemen antrean, pelacakan QR, sinkronisasi *realtime*

Diterima: 06 Juni 2026; Direvisi: 21 Juli 2026; Disetujui: 06 September 2026; Tersedia Online: 22 September 2026; Diterbitkan: 09 Oktober 2026.

1. LATAR BELAKANG

Kualitas layanan perbankan sangat bergantung pada efisiensi sistem antrean. Sistem konvensional yang mengharuskan kehadiran fisik sering memicu penumpukan nasabah dan ketidakpastian waktu layanan. Seiring transformasi digital, diperlukan sistem antrean yang lebih fleksibel, transparan, dan dapat diakses secara daring. Perkembangan komputasi awan memungkinkan layanan ini diakses secara terpusat dan terintegrasi dengan *smartphone* pengguna untuk memperoleh informasi secara efisien(Mude & Mando, 2021). Dalam konteks ini, sinkronisasi data secara real-time menjadi krusial untuk memastikan konsistensi dan kemutakhiran informasi bagi seluruh pengguna.

Selain itu, integrasi teknologi *Quick Response (QR)* Code memberikan kemudahan bagi nasabah untuk melacak perkembangan antrean secara mandiri tanpa harus terus berada di area pelayanan. Namun, sistem antrean digital multi-pengguna menghadapi tantangan besar dalam menjaga konsistensi status layanan. Untuk mencegah transisi status yang tidak sah, diperlukan mekanisme *Finite State Machine* (FSM) yang mampu memodelkan dan memvalidasi perubahan kondisi sistem secara terstruktur dan deterministik (Farhan et al., 2021).

Penerapan FSM telah terbukti berhasil mengendalikan alur operasi pada berbagai sistem. (Krisnandi et al., 2021) dan (Alifudin et al., 2022) menerapkan Finite *State Automata* untuk memastikan validitas transaksi pada *vending machine*. (Peday & Sutejo, 2026) menunjukkan bahwa Deterministic Finite *Automata* (DFA) mampu menghilangkan ambiguitas proses pada sistem bengkel berbasis web, sementara (Azhar et al., 2024) membuktikan bahwa otomasi layanan dapat meningkatkan fleksibilitas akses. Meskipun demikian, mayoritas implementasi tersebut masih berfokus pada perangkat fisik atau belum mengintegrasikan sinkronisasi *real-time* berbasis *cloud* serta pelacakan QR secara terpadu.

Berdasarkan kesenjangan tersebut, penelitian ini mengusulkan *SmartQBank*, yaitu sistem manajemen antrean perbankan berbasis *Finite State Machine* dengan sinkronisasi real-time dan pelacakan QR. FSM digunakan sebagai validator utama untuk transisi status antrean secara deterministik, sementara infrastruktur *cloud* mendistribusikan pembaruan secara instan. Penelitian ini bertujuan merancang, mengimplementasikan, dan mengevaluasi sistem antrean digital yang mampu menjaga integritas data, meningkatkan

transparansi informasi, serta memberikan pengalaman layanan yang efektif dan efisien bagi nasabah maupun pihak perbankan.

2. KAJIAN TEORITIS

Finite State Machine (FSM) merupakan model komputasi abstrak yang digunakan untuk merepresentasikan perilaku sistem melalui sekumpulan *state*, input, dan aturan transisi yang terdefinisi secara formal. FSM memungkinkan sistem berpindah dari satu *state* ke *state* lainnya berdasarkan input atau *event* tertentu sehingga banyak digunakan pada pengembangan perangkat lunak, sistem kendali, dan otomasi industri.

Secara formal, FSM dapat direpresentasikan sebagai himpunan 5-tupel yang terdiri atas *state*, simbol input, fungsi transisi, *state* awal, dan *state* akhir. Pada pendekatan Deterministic Finite *Automata* (DFA), setiap kombinasi *state* dan input hanya memiliki satu kemungkinan transisi yang valid sehingga perilaku sistem dapat diprediksi secara pasti tanpa menimbulkan ambiguitas (Farhan et al., 2021).

Keunggulan FSM terletak pada kemampuannya memodelkan alur kerja secara sistematis, mudah diverifikasi, dan mampu meminimalkan kesalahan logika melalui aturan transisi yang jelas. Pendekatan ini juga memungkinkan validasi terhadap seluruh kemungkinan perpindahan *state* sehingga integritas proses bisnis dapat dipertahankan selama sistem beroperasi.

Penelitian mengenai penerapan *automata* menunjukkan bahwa pendekatan ini efektif dalam mengendalikan alur proses secara terstruktur. (Peday & Sutejo, 2026) menerapkan NFA dan DFA pada sistem informasi bengkel berbasis web dan berhasil mengurangi ambiguitas proses layanan. (Krisnandi et al., 2021) menggunakan Finite *State Automata* pada *vending machine* alat praktik sekolah dan menunjukkan peningkatan keandalan kontrol sistem. Hasil serupa diperoleh (Alifudin et al., 2022) pada *vending machine* tiket pakan hewan, di mana FSM mampu menjaga konsistensi proses transaksi. Berdasarkan penelitian tersebut, FSM dinilai efektif untuk pengendalian alur proses dan validasi status sistem, meskipun integrasinya dengan *cloud computing* dan sinkronisasi data *realtime* masih relatif terbatas.

Cloud computing merupakan paradigma penyediaan sumber daya komputasi melalui internet yang menawarkan skalabilitas, fleksibilitas, dan efisiensi biaya dalam pengembangan sistem digital modern (Alzide, 2024). Perkembangan paradigma ini melahirkan berbagai model layanan, termasuk komputasi serverless yang memungkinkan

sumber daya digunakan secara dinamis sesuai kebutuhan aplikasi sehingga mengurangi kompleksitas operasional sistem (Agarwal et al., 2025).

Salah satu implementasi yang banyak digunakan adalah *Backend-as-a-Service* (BaaS), yaitu layanan yang menyediakan komponen *backend* siap pakai seperti basis data, autentikasi, penyimpanan berkas, dan API. Dengan memanfaatkan BaaS, pengembang dapat lebih berfokus pada pengembangan logika bisnis dan antarmuka pengguna tanpa harus mengelola infrastruktur *backend* secara mandiri.

Penggunaan BaaS mampu mempercepat proses pengembangan, mengurangi kompleksitas pengelolaan server, serta meningkatkan keandalan sistem melalui infrastruktur yang dikelola secara profesional. Integrasi layanan *cloud* dengan basis data relasional juga mendukung pengelolaan data yang lebih konsisten dan mudah diakses melalui internet (Risky Suryaningsih et al., 2026). Selain itu, teknologi *cloud* memungkinkan layanan digital diakses secara fleksibel melalui perangkat bergerak sehingga memperluas jangkauan penggunaan sistem (Mude & Mando, 2021). Pemanfaatan IoT pada mesin layanan mandiri yang terhubung ke internet juga menunjukkan potensi integrasi layanan digital berbasis jaringan untuk meningkatkan efisiensi operasional (Azhar et al., 2024).

Realtime synchronization merupakan mekanisme pertukaran data yang memungkinkan perubahan informasi didistribusikan secara langsung kepada seluruh pengguna yang terhubung tanpa memerlukan pemuatan ulang halaman secara manual. Mekanisme ini menjadi penting pada sistem yang melibatkan banyak pengguna dan membutuhkan konsistensi data secara berkelanjutan.

Pada arsitektur web tradisional, pembaruan data umumnya dilakukan melalui *polling* yang cenderung menghasilkan latensi lebih tinggi dan penggunaan sumber daya jaringan yang kurang efisien. Salah satu solusi yang banyak digunakan adalah *WebSocket* yang memungkinkan komunikasi dua arah secara persisten antara klien dan server. Penelitian (Amirkhanov et al., 2025) menunjukkan bahwa *WebSocket* memberikan performa yang lebih baik dari sisi latensi dan stabilitas dibandingkan komunikasi HTTP konvensional.

Melalui sinkronisasi *realtime*, seluruh pengguna dapat memperoleh informasi yang konsisten dan mutakhir dalam waktu yang hampir bersamaan. Pendekatan ini mendukung

konsep single source of truth sehingga mampu meminimalkan inkonsistensi data serta meningkatkan kualitas layanan sistem (Tiago et al., 2026).

Quick Response (QR) Code merupakan kode matriks dua dimensi yang mampu menyimpan informasi dalam bentuk pola grafis yang dapat dibaca secara cepat menggunakan perangkat pemindai digital. Dibandingkan barcode satu dimensi, QR Code memiliki kapasitas penyimpanan data yang lebih besar dan toleransi yang lebih baik terhadap kerusakan sebagian struktur kode (Siew et al., 2024).

Dalam sistem informasi modern, QR Code banyak digunakan untuk identifikasi, autentikasi, pelacakan informasi, dan akses layanan digital tanpa kontak langsung. Pengguna cukup melakukan pemindaian menggunakan perangkat bergerak untuk memperoleh akses menuju informasi atau layanan tertentu yang terhubung dengan kode tersebut. Pemanfaatan QR Code terbukti mampu meningkatkan efisiensi layanan digital dan mempercepat proses identifikasi pengguna (Zakariya & Arifin, 2025).

Perkembangan teknologi *frontend* mendorong lahirnya berbagai framework yang mendukung pembangunan antarmuka web modern melalui pendekatan komponen yang modular dan pengelolaan *state* yang terstruktur. Salah satu pendekatan yang banyak digunakan adalah *Single Page Application* (SPA), yaitu model aplikasi web yang memungkinkan interaksi pengguna berlangsung tanpa pemuatan ulang halaman secara penuh (Jonathan & Suprihadi, 2023).

Konsep *reactive rendering* memungkinkan antarmuka diperbarui secara otomatis ketika terjadi perubahan data sehingga informasi dapat ditampilkan secara dinamis dan responsif. Kombinasi *framework frontend* modern dengan QR Code dan sinkronisasi *realtime* mendukung pengembangan layanan digital yang interaktif, mudah diakses, dan responsif terhadap kebutuhan pengguna.

Berdasarkan berbagai penelitian yang telah dikaji, dapat disimpulkan bahwa pengembangan sistem antrean digital modern memerlukan integrasi antara mekanisme pengendalian proses, teknologi identifikasi digital, serta penyampaian informasi secara realtime. Kajian sistem antrean pada sektor perbankan menunjukkan bahwa transformasi digital mampu meningkatkan efisiensi layanan, mengurangi waktu tunggu, serta meningkatkan kepuasan pengguna apabila didukung oleh infrastruktur dan integrasi sistem yang memadai (Tesa et al., 2025). Selain itu, pemanfaatan QR Code terbukti mampu mempermudah akses layanan dan pelacakan informasi secara mandiri melalui

perangkat bergerak tanpa memerlukan interaksi langsung dengan petugas layanan (Gaikar et al., 2026). Penelitian mengenai sistem layanan berbasis QR Code juga menunjukkan bahwa digitalisasi proses layanan dapat meningkatkan efisiensi operasional, mempercepat proses transaksi, serta menghasilkan pengelolaan data yang lebih terstruktur dan akurat (Pratama & Mallisza, 2024). Berdasarkan temuan tersebut, penelitian ini mengintegrasikan Finite State Machine sebagai pengendali siklus hidup antrean dengan layanan cloud, sinkronisasi realtime, dan pelacakan berbasis QR Code untuk menghasilkan sistem manajemen antrean perbankan yang lebih terstruktur, responsif, dan mudah diakses oleh pengguna.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental dalam pengembangan sistem manajemen antrean perbankan berbasis web. Fokus utama penelitian adalah merancang dan mengimplementasikan mekanisme pengelolaan antrean yang mampu menjamin konsistensi transisi layanan melalui pendekatan *Finite State Machine* (FSM), serta mendukung sinkronisasi data secara *realtime* pada lingkungan komputasi awan. Tahapan penelitian meliputi pemodelan formal menggunakan konsep *automata*, perancangan arsitektur sistem, perancangan basis data, implementasi sistem, serta pengujian untuk mengevaluasi keandalan dan validitas mekanisme yang dikembangkan.

Pemodelan Formal *Finite State Machine*

Dalam penelitian ini, *Finite State Machine* (FSM) digunakan sebagai mekanisme utama untuk mengendalikan siklus hidup antrean secara terstruktur dan deterministik. Penerapan FSM bertujuan untuk memastikan bahwa setiap perubahan *state* pada entitas antrean hanya dapat terjadi melalui *event* yang telah didefinisikan sebelumnya. Dengan pendekatan tersebut, sistem mampu menjaga konsistensi logika layanan serta mencegah terjadinya perpindahan *state* yang tidak sesuai dengan aturan operasional.

Secara formal, model *automata* yang digunakan pada penelitian ini direpresentasikan menggunakan pendekatan Deterministic Finite *Automata* (DFA). DFA merupakan model komputasi yang memiliki karakteristik deterministik, yaitu setiap kombinasi *state* dan input hanya menghasilkan satu *state* tujuan yang pasti. Menurut (Farhan et al., 2021), DFA dapat dinyatakan dalam bentuk model 5-tupel yang terdiri atas himpunan *state*, himpunan input, fungsi transisi, *state* awal, dan himpunan *state* akhir. Pendekatan DFA banyak digunakan dalam pemodelan sistem karena mampu

menghilangkan ambiguitas proses dan menjaga konsistensi alur operasional (Singh & Goyal, 2022). Model tersebut dinyatakan pada Persamaan (1).

$$M = (Q, \Sigma, \delta, S, F) \quad (1)$$

Berdasarkan implementasi yang digunakan pada sistem *SmartQBank*, setiap komponen dalam model 5-tupel tersebut didefinisikan sebagai berikut.

Q merupakan himpunan berhingga dari *state* yang merepresentasikan tahapan layanan antrean, yaitu:

Q = {IDLE, WAITING, CALLED, SERVING, NO_SHOW, COMPLETED, CANCELLED, TRANSFERRED}

Σ merupakan himpunan *event* yang dapat memicu perpindahan *state* pada sistem, yaitu:

Σ = {ENQUEUE, CALL_NEXT, CUSTOMER_ARRIVED, TIMEOUT, REQUEUE, COMPLETE, CANCEL, TRANSFER}

δ merupakan fungsi transisi yang memetakan pasangan *state* dan *event* menuju *state* berikutnya.

$$\delta : Q \times \Sigma \rightarrow Q$$

S merupakan *state* awal yang menunjukkan kondisi token sebelum memasuki siklus layanan, yaitu:

$$S = IDLE$$

F merupakan himpunan *state* akhir yang menandakan bahwa proses layanan telah selesai dan tidak dapat melakukan transisi lanjutan, yaitu:

$$F = \{COMPLETED, CANCELLED\}$$

Fungsi transisi δ menjadi komponen utama yang mengatur perpindahan *state* pada sistem antrean. Setiap *event* yang diterima akan dievaluasi berdasarkan *state* saat ini untuk menentukan apakah transisi yang diminta valid atau tidak. Pemetaan fungsi transisi yang digunakan dalam penelitian ini ditunjukkan pada Tabel 1.

Tabel 1. Fungsi Transisi *Finite State Machine SmartQBank*

State Saat Ini	*Event*	*State* Berikutnya
IDLE	ENQUEUE	WAITING
WAITING	CALL_NEXT	CALLED
WAITING	CANCEL	CANCELLED
CALLED	CUSTOMER_ARRIVED	SERVING
CALLED	TIMEOUT	NO_SHOW
NO_SHOW	REQUEUE	WAITING

NO_SHOW	CANCEL	CANCELLED
SERVING	COMPLETE	COMPLETED
SERVING	TRANSFER	TRANSFERRED
TRANSFERRED	ENQUEUE	WAITING

Tabel 1 menunjukkan fungsi transisi yang mengatur seluruh siklus hidup token antrean pada sistem *SmartQBank*. Setiap *event* yang diterima akan diproses berdasarkan *state* saat ini sehingga perpindahan *state* hanya dapat terjadi apabila memenuhi aturan transisi yang telah ditentukan sebelumnya. Dalam konteks operasional sistem antrean, *event* ENQUEUE digunakan untuk memasukkan token ke dalam antrean sehingga *state* berubah dari IDLE menjadi WAITING. Ketika operator melakukan pemanggilan, *event* CALL_NEXT akan mengubah *state* menjadi CALLED. Selanjutnya, apabila pelanggan hadir untuk menerima layanan, *event* CUSTOMER_ARRIVED akan mengubah *state* menjadi SERVING. Sebaliknya, apabila pelanggan tidak memberikan respons dalam batas waktu yang ditentukan, *event* TIMEOUT akan memindahkan token ke *state* NO_SHOW.

Pada *state* NO_SHOW, pelanggan diberikan opsi untuk kembali ke antrean melalui *event* REQUEUE atau membatalkan antrean melalui *event* CANCEL. Setelah proses pelayanan berlangsung, *event* COMPLETE digunakan untuk mengakhiri layanan sehingga token memasuki *state* COMPLETED. Selain itu, sistem juga menyediakan mekanisme pengalihan layanan menggunakan *event* TRANSFER yang akan memindahkan token ke *state* TRANSFERRED sebelum dimasukkan kembali ke antrean melalui *event* ENQUEUE.

Implementasi FSM pada penelitian ini menerapkan pendekatan deterministik sehingga setiap kombinasi *state* dan *event* hanya memiliki satu hasil transisi yang valid. Apabila sistem menerima *event* yang tidak terdefinisi pada fungsi transisi untuk *state* tertentu, maka *event* tersebut dianggap tidak valid dan tidak menghasilkan perubahan *state*. Mekanisme ini berfungsi sebagai lapisan validasi yang menjaga integritas data serta mencegah terjadinya pelanggaran alur layanan yang telah ditentukan. Pendekatan serupa juga diterapkan pada berbagai penelitian berbasis DFA untuk memastikan bahwa seluruh proses operasional berjalan sesuai aturan transisi yang telah didefinisikan sebelumnya (Krisnandi et al., 2021)

Perancangan Sistem *SmartQBank*

Perancangan sistem *SmartQBank* dilakukan dengan mengintegrasikan konsep *Finite State Machine* (FSM), komputasi awan, basis data relasional, dan sinkronisasi data secara *realtime* ke dalam sebuah sistem manajemen antrean berbasis web. Sistem dirancang untuk mendukung proses layanan perbankan yang terstruktur, mudah diakses, dan mampu memberikan informasi antrean secara aktual kepada seluruh pengguna yang terhubung. Arsitektur yang dikembangkan memisahkan komponen antarmuka pengguna, logika bisnis, layanan *backend*, serta mekanisme sinkronisasi data sehingga sistem dapat dikelola dan dikembangkan secara modular.

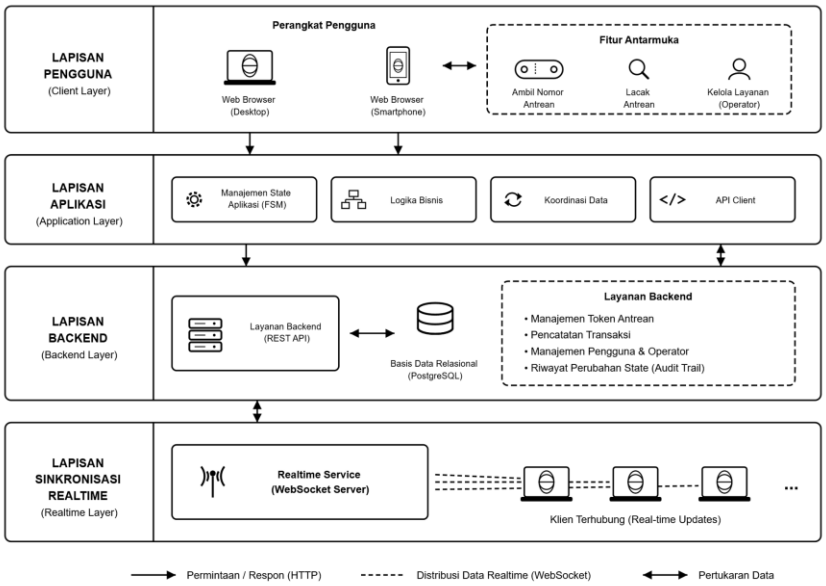
Arsitektur Sistem

Arsitektur *SmartQBank* menggunakan pendekatan multi-layer (lapisan pengguna, aplikasi, *backend*, dan sinkronisasi *realtime*) guna memisahkan tanggung jawab komponen agar sistem lebih terstruktur, mudah dipelihara, serta skalabel (Ye & Zhang, 2022).

Pada lapisan pengguna, sistem diakses via peramban web di komputer maupun smartphone untuk layanan nomor dan pelacakan antrean. Permintaan ini diteruskan ke lapisan aplikasi yang bertugas menangani logika bisnis dan menyajikan antarmuka responsif melalui pendekatan pengembangan aplikasi web modern berbasis komponen (Risky Suryaningsih et al., 2026).

Sebagai pusat pengelolaan, lapisan *backend* menyimpan data transaksi menggunakan basis data relasional untuk menjaga integritas informasi. Lapisan ini juga menyediakan antarmuka pemrograman aplikasi (API) guna memfasilitasi pertukaran data secara terpusat.

Untuk mendistribusikan pembaruan informasi secara langsung tanpa perlu memuat ulang halaman, sistem menggunakan mekanisme komunikasi *realtime* berbasis *WebSocket*. Hal ini terbukti efisien dalam menjaga konsistensi data pada lingkungan dengan banyak pengguna secara bersamaan (Amirkhanov et al., 2025). Struktur arsitektur ini ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur Sistem *SmartQBank*

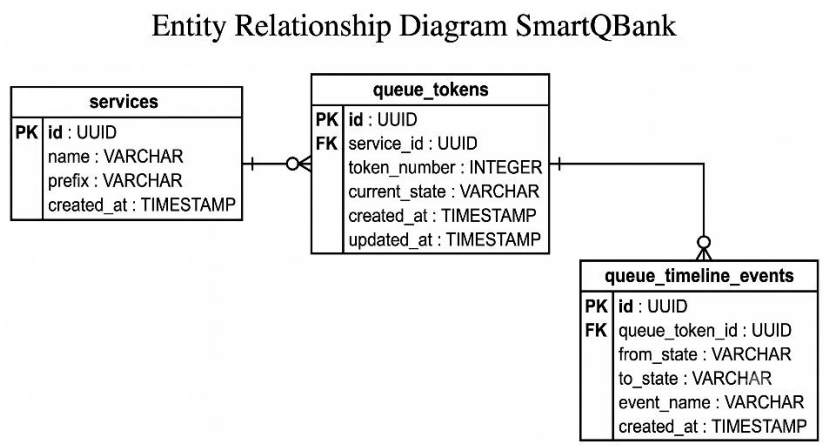
Perancangan Basis Data

Basis data pada penelitian ini dirancang menggunakan model relasional untuk menjamin integritas, konsistensi, dan kemudahan pengelolaan data. Model relasional dipilih karena mampu mendukung pengelolaan data yang terstruktur melalui hubungan antarentitas serta menjamin konsistensi transaksi yang terjadi selama proses operasional sistem berlangsung. Selain itu, model ini banyak digunakan dalam pengembangan sistem informasi modern karena mampu menyediakan mekanisme penyimpanan data yang andal dan mudah dipelihara (Risky Suryaningsih et al., 2026).

Perancangan basis data dilakukan dengan mengidentifikasi entitas utama yang terlibat dalam proses layanan antrean beserta hubungan antarentitasnya. Setiap entitas direpresentasikan dalam bentuk tabel yang saling terhubung melalui relasi tertentu sehingga memungkinkan pengelolaan data dilakukan secara terstruktur, efisien, dan konsisten. Integrasi basis data relasional dengan lingkungan komputasi awan juga mendukung akses data yang lebih fleksibel serta mendukung kebutuhan skalabilitas sistem pada lingkungan layanan digital modern (Alzide, 2024)

Selain mendukung penyimpanan data operasional, basis data juga dirancang untuk menyediakan jejak historis (audit trail) terhadap setiap perubahan yang terjadi selama proses layanan. Mekanisme ini memungkinkan seluruh aktivitas sistem terdokumentasi dengan baik sehingga dapat digunakan untuk kebutuhan monitoring, evaluasi layanan,

maupun analisis operasional di masa mendatang. Penyimpanan data yang terstruktur dan terdokumentasi secara historis merupakan salah satu karakteristik penting dalam sistem berbasis layanan yang memanfaatkan arsitektur *cloud* modern (Agarwal et al., 2025). Rancangan hubungan antarentitas pada sistem ditunjukkan pada Gambar 2.



Gambar 2. Entity Relationship Diagram (ERD) *SmartQBank*

Alur Operasional Sistem

Alur operasional sistem menggambarkan urutan proses yang terjadi sejak pengguna mengambil nomor antrean hingga layanan selesai diberikan. Proses ini dirancang berdasarkan aturan transisi yang telah didefinisikan pada model *Finite State Machine* sehingga setiap tahapan layanan dapat berlangsung secara terstruktur, konsisten, dan sesuai dengan prosedur layanan yang telah ditetapkan.

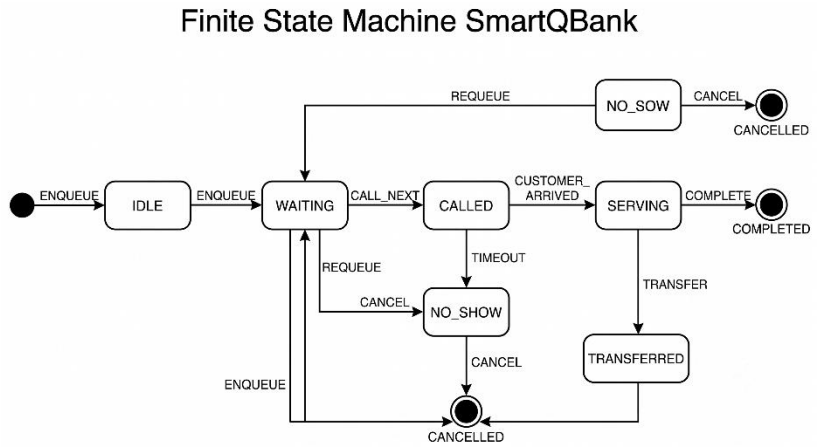
Proses dimulai ketika pengguna melakukan pengambilan nomor antrean melalui antarmuka sistem. Setelah data antrean berhasil direkam, sistem menghasilkan identitas antrean yang dapat digunakan untuk memantau perkembangan layanan. Identitas tersebut selanjutnya dihubungkan dengan mekanisme pelacakan berbasis QR Code sehingga pengguna dapat mengakses informasi antrean secara mandiri melalui perangkat bergerak tanpa harus melakukan interaksi langsung dengan petugas layanan. Pemanfaatan QR Code sebagai media akses layanan digital terbukti mampu meningkatkan efisiensi proses pelacakan informasi dan memberikan kemudahan bagi pengguna dalam memperoleh informasi secara *realtime* (Siew et al., 2024).

Selanjutnya, operator dapat melakukan berbagai tindakan layanan yang akan memicu perubahan *state* sesuai dengan aturan transisi yang telah ditentukan. Setiap perubahan *state* yang terjadi akan disimpan ke dalam basis data dan secara otomatis

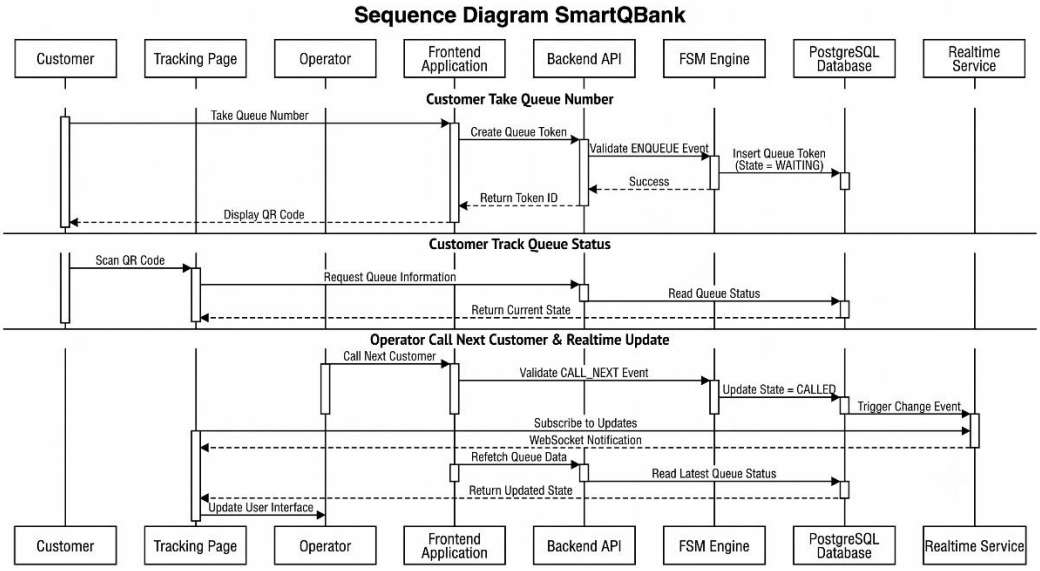
didistribusikan kepada seluruh pengguna yang sedang terhubung melalui mekanisme sinkronisasi *realtime*. Pendekatan ini memungkinkan seluruh pengguna memperoleh informasi yang sama pada waktu yang hampir bersamaan sehingga risiko terjadinya perbedaan informasi dapat diminimalkan. Komunikasi *realtime* berbasis *WebSocket* terbukti mampu mendukung distribusi data secara cepat dan stabil pada lingkungan sistem yang melibatkan banyak pengguna secara bersamaan (Amirkhanov et al., 2025).

Mekanisme sinkronisasi otomatis tersebut mendukung terciptanya proses layanan yang lebih responsif dan terintegrasi karena setiap perubahan informasi dapat langsung diterima oleh seluruh pihak yang berkepentingan. Penerapan otomatisasi *realtime* pada sistem informasi modern juga berkontribusi terhadap peningkatan efisiensi operasional serta percepatan penyampaian informasi kepada pengguna (Tiago et al., 2026)

Urutan interaksi antara pengguna, sistem aplikasi, layanan *backend*, basis data, dan mekanisme sinkronisasi *realtime* divisualisasikan melalui diagram urutan (sequence diagram) yang ditunjukkan pada Gambar 4. Sementara itu, representasi visual dari mekanisme transisi *state* yang menjadi inti pengendalian layanan ditunjukkan pada Gambar 3.



Gambar 3. Diagram *Finite State Machine SmartQBank*

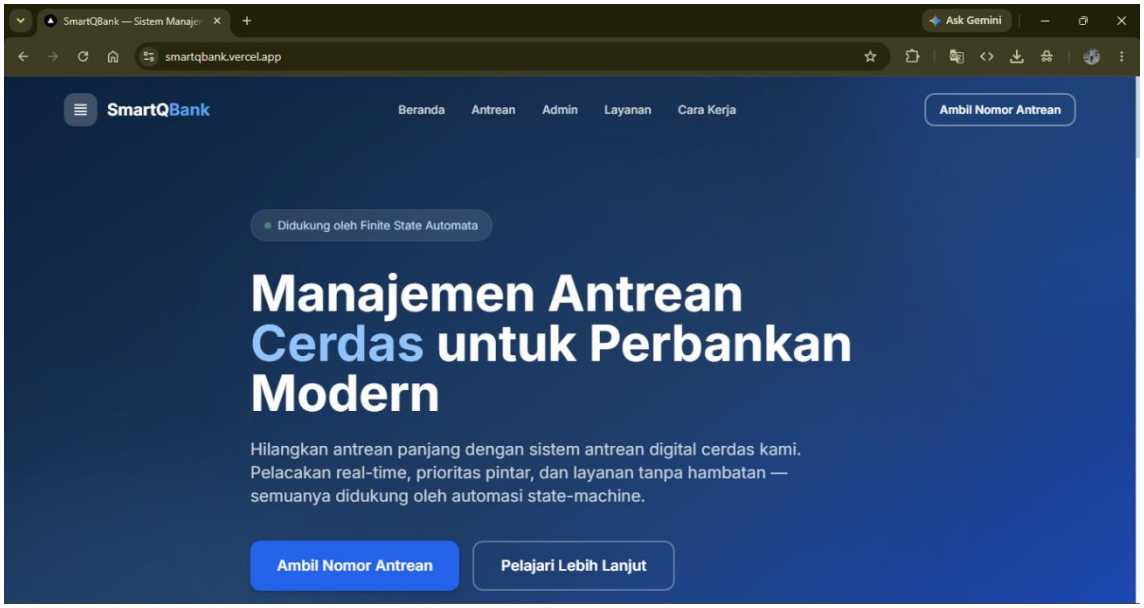


Gambar 4. Sequence Diagram Operasional *SmartQBank*

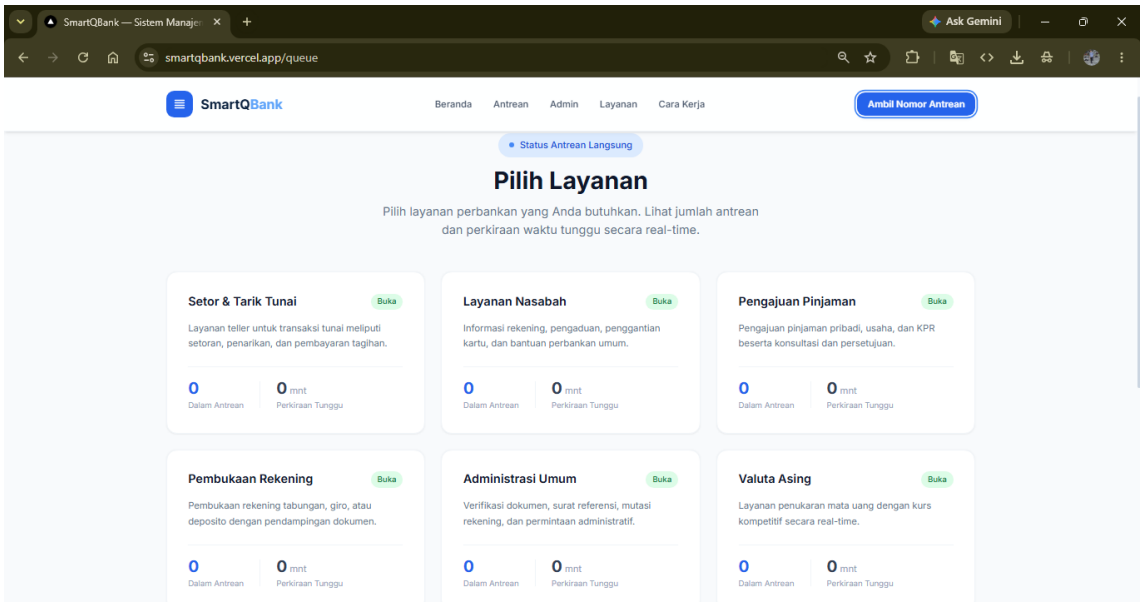
4. HASIL DAN PEMBAHASAN

Sistem SmartQBank telah berhasil diimplementasikan dan dapat diakses melalui alamat:

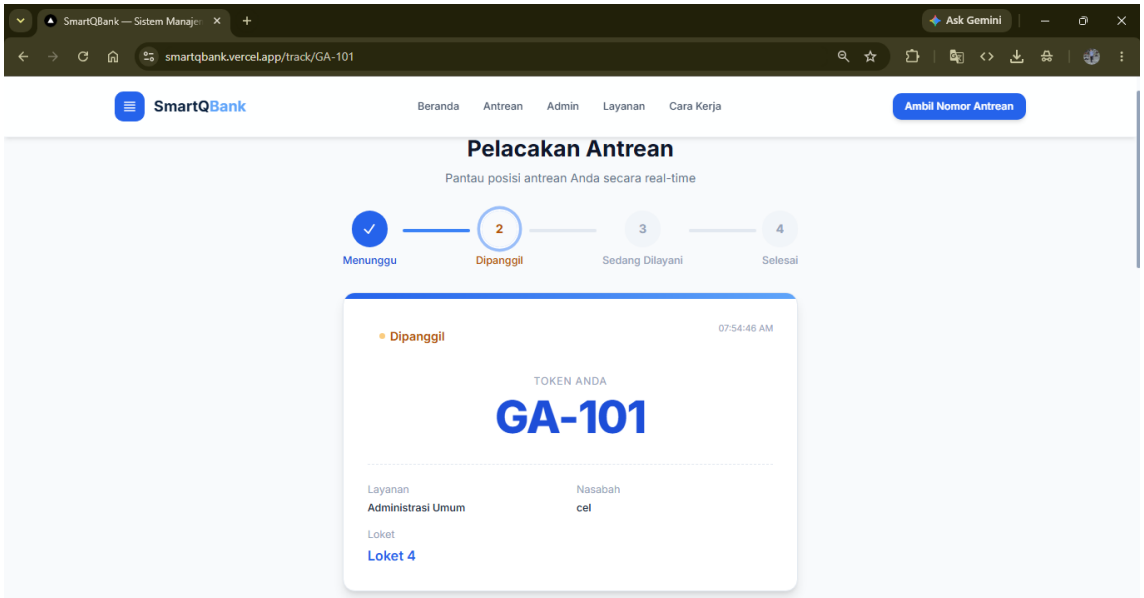
https://smartqbank.vercel.app/



Gambar 5. Halaman Antarmuka *SmartQBank*



Gambar 6. Antarmuka Pemilihan Layanan Antrean



Gambar 7. Antarmuka Pengambilan Nomor Antrean

Implementasi ini memungkinkan pengguna melakukan pengambilan nomor antrean, pemantauan status layanan secara realtime, serta pengelolaan antrean oleh operator melalui antarmuka berbasis web. Implementasi *SmartQBank* menghasilkan sebuah sistem manajemen antrean berbasis web yang mengintegrasikan mekanisme *Finite State Machine* (FSM), penyimpanan data berbasis *cloud*, sinkronisasi *realtime*, dan pelacakan antrean melalui Quick Response (QR) Code. Pengujian dilakukan untuk mengevaluasi keakuratan fungsi transisi FSM, kemampuan sistem dalam menangani

transisi tidak valid, serta keandalan integrasi antar komponen yang terlibat dalam proses layanan.

Secara umum, pengujian difokuskan pada dua aspek utama, yaitu validasi model FSM sebagai pengendali siklus hidup antrean dan evaluasi integrasi sistem dalam lingkungan komputasi terdistribusi. Hasil pengujian digunakan untuk memastikan bahwa seluruh proses layanan berjalan sesuai dengan aturan yang telah dirancang pada tahap pemodelan dan perancangan sistem.

Pengujian *Finite State Machine*

Pengujian *Finite State Machine* dilakukan untuk memastikan bahwa setiap *event* yang diberikan kepada sistem menghasilkan transisi *state* yang sesuai dengan fungsi transisi yang telah didefinisikan pada tahap pemodelan formal. Selain itu, pengujian ini juga bertujuan untuk memverifikasi kemampuan sistem dalam menolak *event* yang tidak valid sehingga integritas siklus layanan dapat dipertahankan.

Pengujian Transisi Valid

Pengujian transisi valid dilakukan dengan memberikan *event* yang sesuai terhadap *state* tertentu berdasarkan aturan transisi yang telah ditetapkan. Setiap skenario diuji untuk memastikan bahwa sistem menghasilkan *state* tujuan yang benar.

Tabel 2. Hasil Pengujian Transisi Valid *Finite State Machine*

***State* Awal**	***Event***	***State* Hasil**	**Hasil Pengujian**
IDLE	ENQUEUE	WAITING	Berhasil
WAITING	CALL_NEXT	CALLED	Berhasil
WAITING	CANCEL	CANCELLED	Berhasil
CALLED	CUSTOMER_ARRIVED	SERVING	Berhasil
CALLED	TIMEOUT	NO_SHOW	Berhasil
NO_SHOW	REQUEUE	WAITING	Berhasil
NO_SHOW	CANCEL	CANCELLED	Berhasil
SERVING	COMPLETE	COMPLETED	Berhasil
SERVING	TRANSFER	TRANSFERRED	Berhasil
TRANSFERRED	ENQUEUE	WAITING	Berhasil

Berdasarkan hasil pengujian pada Tabel 2, seluruh skenario transisi yang dirancang berhasil dieksekusi sesuai dengan fungsi transisi yang telah didefinisikan pada model *Finite State Machine*. Tidak ditemukan kondisi di mana sistem menghasilkan *state* yang berbeda dari *state* tujuan yang telah ditentukan. Hasil tersebut menunjukkan bahwa implementasi transition function pada *SmartQBank* mampu menerjemahkan model formal *automata* ke dalam mekanisme operasional sistem secara konsisten.

Dari sudut pandang teori *automata*, keberhasilan seluruh transisi valid menunjukkan bahwa model deterministic finite *automata* yang digunakan telah memenuhi karakteristik deterministik, yaitu setiap kombinasi *state* dan *event* hanya menghasilkan satu *state* tujuan yang pasti. Kondisi ini sangat penting dalam sistem layanan perbankan karena menghilangkan ambiguitas proses dan mencegah terjadinya konflik status antar pengguna yang mengakses sistem secara bersamaan.

Hasil penelitian ini sejalan dengan penelitian (Peday & Sutejo, 2026) yang menyatakan bahwa pendekatan DFA mampu menghilangkan ketidakpastian alur layanan melalui mekanisme validasi transisi yang terstruktur. Temuan serupa juga dilaporkan oleh (Krisnandi et al., 2021) dan (Alifudin et al., 2022) yang memanfaatkan Finite *State Automata* sebagai pengendali logika operasional pada sistem *vending machine*. Pada kedua penelitian tersebut, keberhasilan sistem ditentukan oleh kemampuan *automata* dalam memastikan bahwa setiap proses hanya dapat berlangsung melalui jalur transisi yang sah.

Dengan demikian, implementasi FSM pada *SmartQBank* tidak hanya berfungsi sebagai model konseptual, tetapi juga sebagai mekanisme validasi yang secara langsung menjaga integritas data dan konsistensi layanan selama sistem beroperasi.

Pengujian Transisi Tidak Valid

Selain menguji transisi yang sah, pengujian juga dilakukan terhadap sejumlah *event* yang tidak didefinisikan pada fungsi transisi. Tujuan pengujian ini adalah memastikan bahwa sistem mampu menolak instruksi yang berpotensi menyebabkan inkonsistensi data maupun pelanggaran alur layanan.

Tabel 3. Hasil Pengujian Transisi Tidak Valid *Finite State Machine*

***State* Awal**	***Event* Tidak Valid**	**Hasil yang Diharapkan**	**Hasil Pengujian**
WAITING	COMPLETE	Ditolak	Berhasil
WAITING	TRANSFER	Ditolak	Berhasil
COMPLETED	CALL_NEXT	Ditolak	Berhasil
CANCELLED	COMPLETE	Ditolak	Berhasil
NO_SHOW	COMPLETE	Ditolak	Berhasil

Hasil pengujian pada Tabel 3 menunjukkan bahwa seluruh *event* yang tidak didefinisikan pada fungsi transisi berhasil ditolak oleh sistem. Ketika *event* tidak valid diberikan, tidak terjadi perubahan *state* maupun mutasi data pada basis data. Dengan kata

lain, token antrean tetap berada pada *state* asal sehingga siklus layanan tidak mengalami penyimpangan.

Keberhasilan mekanisme penolakan ini menunjukkan bahwa FSM pada *SmartQBank* beroperasi sebagai lapisan validasi utama sebelum perubahan data diterapkan ke sistem. Implementasi tersebut sejalan dengan konsep *deterministic finite automata* yang hanya mengizinkan transisi yang telah didefinisikan secara formal pada fungsi transisinya. *Event* yang berada di luar definisi fungsi transisi dianggap tidak valid sehingga tidak dapat menghasilkan perpindahan *state*.

Temuan ini mendukung hasil penelitian (Farhan et al., 2021) yang menunjukkan bahwa penerapan model *automata* mampu digunakan sebagai mekanisme validasi logika untuk mencegah kesalahan proses pada aplikasi berbasis *state*. Selain itu, hasil ini juga memperkuat argumentasi (Peday & Sutejo, 2026) bahwa validasi berbasis DFA efektif untuk mengurangi anomali operasional yang disebabkan oleh kesalahan pengguna maupun ketidaksesuaian prosedur layanan.

Implikasi praktis dari kemampuan tersebut adalah meningkatnya keandalan sistem dalam lingkungan layanan perbankan yang melibatkan banyak pengguna dan aktivitas secara bersamaan. Dengan adanya mekanisme proteksi terhadap transisi tidak valid, risiko inkonsistensi data dan pelanggaran prosedur layanan dapat diminimalkan secara signifikan.

Pengujian Integrasi Sistem

Pengujian integrasi dilakukan untuk mengevaluasi interaksi antar komponen yang membentuk *SmartQBank*. Pengujian mencakup proses penerbitan token antrean, penyimpanan data, pelacakan antrean melalui QR Code, sinkronisasi *realtime*, serta pembaruan informasi pada antarmuka pengguna.

Tabel 4. Hasil Pengujian Integrasi Sistem

Komponen	Skenario Pengujian	Hasil
Penerbitan Token	Pembuatan nomor antrean baru	Berhasil
Penyimpanan Data	Penyimpanan data antrean ke basis data	Berhasil
QR Code	Akses pelacakan antrean melalui QR Code	Berhasil
Dashboard Operator	Pengelolaan layanan melalui antarmuka admin	Berhasil

Sinkronisasi *Realtime*	Distribusi perubahan data ke seluruh klien	Berhasil
Pemulihan Data	Pemuatan ulang halaman dan sinkronisasi ulang data	Berhasil

Berdasarkan hasil pengujian integrasi pada Tabel 4, seluruh komponen yang membentuk *SmartQBank* mampu beroperasi sesuai dengan rancangan sistem yang telah ditetapkan. Proses penerbitan token antrean, penyimpanan data, pelacakan antrean melalui QR Code, sinkronisasi *realtime*, serta pemulihan data berhasil dijalankan tanpa ditemukan kegagalan fungsi selama proses pengujian. Hasil tersebut menunjukkan bahwa integrasi antara komponen frontend, *backend*, basis data, dan layanan sinkronisasi *realtime* mampu mendukung operasional sistem secara terpadu.

Keberhasilan integrasi tersebut menunjukkan bahwa pendekatan arsitektur berbasis *cloud* yang digunakan mampu mendukung pertukaran data secara konsisten antar komponen sistem. Pemanfaatan layanan *cloud* memungkinkan proses penyimpanan, pengelolaan, dan distribusi data dilakukan secara terpusat sehingga setiap perubahan informasi dapat diakses oleh seluruh pengguna yang terhubung. Temuan ini sejalan dengan (Alzide, 2024) yang menyatakan bahwa *cloud computing* memberikan fleksibilitas, skalabilitas, dan kemudahan integrasi dalam pengembangan sistem informasi modern. Selain itu, pendekatan layanan *backend* berbasis *cloud* juga mendukung efisiensi pengelolaan sumber daya dan penyederhanaan proses pengembangan sistem sebagaimana dijelaskan oleh (Agarwal et al., 2025).

Dari aspek layanan pengguna, fitur pelacakan antrean melalui QR Code berhasil memberikan akses informasi secara mandiri tanpa memerlukan interaksi langsung dengan petugas layanan. Pengguna dapat memantau perkembangan antrean melalui perangkat bergerak yang terhubung ke sistem. Hasil ini mendukung penelitian (Siew et al., 2024) yang menunjukkan bahwa integrasi teknologi QR Code pada sistem berbasis web mampu meningkatkan kemudahan akses informasi, efisiensi layanan, dan pengalaman pengguna dalam memperoleh informasi secara *realtime*.

Pada aspek sinkronisasi data, mekanisme *realtime* berhasil mendistribusikan perubahan informasi kepada seluruh klien yang sedang terhubung tanpa memerlukan proses pemuatan ulang halaman secara manual. Setiap perubahan *state* yang dilakukan oleh operator dapat segera diterima oleh pengguna lain dalam waktu yang hampir bersamaan. Temuan ini sejalan dengan penelitian (Amirkhanov et al., 2025) yang

menyatakan bahwa komunikasi berbasis *WebSocket* mampu menyediakan distribusi data dengan latensi rendah dan konsistensi informasi yang lebih baik dibandingkan pendekatan komunikasi tradisional berbasis permintaan berkala (polling). Selain itu, keberhasilan sinkronisasi *realtime* pada *SmartQBank* juga menunjukkan efektivitas penerapan otomatisasi informasi secara berkelanjutan sebagaimana dilaporkan oleh (Tiago et al., 2026)

Hasil penelitian ini juga memiliki keterkaitan dengan penelitian (Azhar et al., 2024) yang menunjukkan bahwa integrasi layanan berbasis jaringan mampu meningkatkan efisiensi penyampaian informasi kepada pengguna. Meskipun penelitian tersebut berfokus pada layanan internet berbasis perangkat IoT, keduanya memiliki kesamaan dalam pemanfaatan komunikasi jaringan sebagai media distribusi layanan secara otomatis dan berkelanjutan. Selain itu, hasil penelitian ini mendukung konsep sistem informasi terintegrasi yang dikemukakan oleh (Mude & Mando, 2021) yaitu bahwa layanan berbasis jaringan memungkinkan pengguna memperoleh informasi secara fleksibel melalui perangkat yang terhubung ke internet.

Secara keseluruhan, hasil pengujian integrasi menunjukkan bahwa *SmartQBank* berhasil menggabungkan mekanisme *Finite State Machine*, basis data relasional, layanan *cloud*, sinkronisasi *realtime*, dan pelacakan antrean berbasis QR Code ke dalam satu ekosistem layanan yang terintegrasi. Integrasi tersebut memungkinkan sistem menjaga konsistensi informasi, meningkatkan transparansi proses layanan, serta memberikan pengalaman pengguna yang lebih responsif. Temuan ini menunjukkan bahwa pendekatan yang diusulkan memiliki potensi untuk diterapkan pada berbagai layanan publik yang membutuhkan pengelolaan antrean secara efisien, terukur, dan berbasis teknologi digital modern.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan mengimplementasikan *SmartQBank* sebagai sistem manajemen antrean perbankan berbasis *Finite State Machine* (FSM) yang terintegrasi dengan teknologi *cloud* computing, sinkronisasi *realtime*, dan pelacakan antrean menggunakan Quick Response (QR) Code. Penerapan FSM sebagai mekanisme inti pengendalian layanan mampu memastikan bahwa setiap perubahan *state* antrean berlangsung secara deterministik dan sesuai dengan aturan transisi yang telah ditetapkan. Integrasi antara model *automata*t, basis data relasional, dan komunikasi *realtime*

menghasilkan sistem yang mampu menjaga konsistensi data serta menyediakan informasi antrean secara aktual kepada seluruh pengguna yang terhubung.

Berdasarkan hasil pengujian yang dilakukan, sistem mampu menjalankan seluruh transisi valid sesuai dengan model formal yang dirancang serta berhasil menolak transisi yang tidak sah. Selain itu, integrasi antar komponen sistem berjalan dengan baik sehingga proses penerbitan token, penyimpanan data, sinkronisasi informasi, dan pelacakan antrean dapat dilakukan secara efektif. Temuan ini menunjukkan bahwa pendekatan *Finite State Machine* dapat diterapkan secara praktis untuk meningkatkan keandalan dan keteraturan layanan antrean pada lingkungan perbankan modern.

Penelitian ini masih memiliki keterbatasan karena pengujian dilakukan pada lingkungan pengembangan dan belum melibatkan implementasi langsung pada institusi perbankan dalam skala operasional yang sesungguhnya. Oleh karena itu, penelitian selanjutnya dapat difokuskan pada pengujian performa sistem dalam skenario pengguna yang lebih besar, analisis beban layanan secara simultan, serta integrasi dengan teknologi kecerdasan buatan untuk melakukan prediksi waktu tunggu dan optimasi distribusi layanan antrean secara otomatis.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi Ilmu Komputer Universitas Negeri Medan yang telah memberikan dukungan akademik selama proses penelitian dan penyusunan artikel ini. Penulis juga menyampaikan apresiasi kepada seluruh pihak yang telah memberikan masukan, bantuan, dan dukungan selama proses pengembangan sistem *SmartQBank* hingga penelitian ini dapat diselesaikan dengan baik.

DAFTAR REFERENSI

Agarwal, S., Rodriguez, M. A., & Buyya, R. (2025). Dynamic Function Configuration and its Management in Serverless Computing: A Taxonomy and Future Directions. In *Journal of the ACM: XX*. https://doi.org/XXXXXXX.XXXXXXX

Alifudin, M. I., Gata, W., Bayhaqy, A., Hermaliani, E. H., Putra, J. L., & Studi, P. (2022). *Penerapan Finite State Automata Pada Desain Vending Machine Tiket Pakan Hewan Kebun Binatang*.

Alzide, S. (2024). Cloud Computing: Evolution, Challenges, and Future Prospects. *Journal of Information Technology, Cybersecurity, and Artificial Intelligence*, *1*(1), 52–63. https://doi.org/10.70715/jitcai.2024.v1.i1.007

Amirkhanov, B., Amirkhanova, G., Kunelbayev, M., Adilzhanova, S., & Tokhtassyn, M. (2025). Evaluating HTTP, MQTT over TCP and MQTT over WEBSOCKET for digital twin applications: A comparative analysis on latency, stability, and

integration. *International Journal of Innovative Research and Scientific Studies*, *8*(1), 679–694. https://doi.org/10.53894/ijirss.v8i1.4414

Azhar, R., Santoso, H., Putu Hariyadi, I., & Afriandy, M. (2024). Prototype Mesin Internet dengan Pembayaran Koin Menggunakan Mikrokontroller Rock Pi Internet Machine Prototype with a Coin Payment Using a Rock Pi Microcontroller. *Jurnal Bumigora Information Technology (BITe)*, *6*(1), 59–72. https://doi.org/10.30812/bite/v6i1.3534

Farhan, S., Fitri, I., Infoman, F., Studi Informatika, P., Sawo Manila Pasar Minggu, J., & Selatan, J. (2021). *Implementasi Transisi Finite State Automata dengan Aplikasi Mesin Abstrak DFA dan NFA Berbasis Android*.

Gaikar, A., Ambre, K., Sarkavas, R., Kumari, S., Jadhav, S., & Professor, A. (2026). A Study on JOYFOOD: QR Code Based Food Ordering System for Restaurants. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *14*, 2321–9653. www.ijraset.com

Peday, A. I. W. C. R., & Sutejo, H. (2026). Implementasi NFA dan DFA untuk pemodelan alur layanan pada sistem informasi bengkel berbasis web. *Journal of Informatics, Intelligent Technology, Cybersecurity and Software (JIITeCS) (Pada data Anda hanya tertulis "Journal of Informatics")*, *01*(1), 49–54. https://doi.org/10.31315/jiitecs.vxxix.xxx

Jonathan, R., & Suprihadi. (2023). Development of Front-End Web Applications Utilizing Single Page Application Framework and React.js Library. *International Journal Software Engineering and Computer Science (IJSECS)*, *3*(3), 529–536. https://doi.org/10.35870/ijsecs.v3i3.1943

Krisnandi, D., Dinul Fatiha, Z., Putra, J. L., Saputra, S. A., & Gata, W. (2021). KONSEP FINITE STATE AUTOMATA PADA DESAIN VENDING MACHINE ALAT PRAKTIK DI SEKOLAH MENENGAH KEJURUAN INFO ARTIKEL ABSTRAK. *Jurnal Explore IT (Jurnal Keilmuan dan Aplikasi Teknik Informatika)*, *12*(1), 22–27. https://doi.org/10.35891/explorit.v13i1.2568

Mude, A., & Mando, L. B. F. (2021). Implementasi Keamanan Rumah Cerdas Menggunakan Internet of Things dan Biometric Sistem. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *21*(1), 179–188. https://doi.org/10.30812/matrik.v21i1.1381

Pratama, M., & Mallisza, D. (2024). JURNAL MANAJEMEN TEKNOLOGI INFORMATIKA RANCANG BANGUN SISTEM PEMESANAN MENU MAKANAN PADA RUMAH MAKAN CHANIA DENGAN QR CODE BERBASIS WEB. *Jurnal Manajemen Teknologi Informatika (JENTIK)*, *2*(3), 25115.

Risky Suryaningsih, Sulaibatul Aslamiyah, & Tintin Harlina. (2026). Pengembangan Aplikasi Web Terpadu untuk Pendaftaran dan Pemeriksaan Pasien Menggunakan Agile Scrum di Puskesmas Kabat. *JEKIN - Jurnal Teknik Informatika*, *6*(2), 20–34. https://doi.org/10.58794/jekin.v6i2.2102

Siew, E. S. K., Chong, Z. Y., Sze, S. N., & Hardi, R. (2024). Streamlining Attendance Management in Education: A Web-Based System Combining Facial Recognition and QR Code Technology. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *33*(2), 198–208. https://doi.org/10.37934/araset.33.2.198208

Singh, R., & Goyal, Dr. G. (2022). Algorithm Design for Deterministic Finite Automata for a Given Regular Language with Prefix Strings. *Journal of Scientific Research*, *66*(02), 16–21. https://doi.org/10.37398/jsr.2022.660203

Tesa, C. A., Haris, M. S., & Kusuma, W. T. (2025). An Analysis of Digital Queuing Systems in Indonesian Banking Services: A Systematic Literature Review. *International Journal Software Engineering and Computer Science (IJSECS)*, *5*(2), 724–732. https://doi.org/10.35870/ijsecs.v5i2.4379

Tiago, C., Gon鏘lo, R., Lio, G., Vasco, A., & V韙or, F. (2026). Intelligent Real-Time Kanban Automation Using Ultra-Wideband Positioning: Methodologies and Performance Evaluation. *Intelligent and Sustainable Manufacturing*, *3*(1), 10006–10006. https://doi.org/10.70322/ism.2026.10006

Ye, K., & Zhang, L.-J. (2022). *Cloud computing – CLOUD 2021: 14th international conference held as part of the Services Conference Federation, SCF 2021 virtual event, December 10–14, 2021 proceedings (Lecture Notes in Computer Science 12989)*. Springer Nature Switzerland AG. https://doi.org/https://doi.org/10.1007/978-3-030-96326-2

Zakariya, A., & Arifin, Ah. A. (2025). The Application of Digital Payments (QRIS & E-Wallet) on the Efficiency of MSME Transactions. *Journal of Finance, Economics and Business*, *4*(2), 1–19. https://doi.org/10.59827/jfeb.v4i2.326