Perancangan Dan Implementasi Backend Menggunakan Supabase Pada Sistem Smart Shopping Cart Berbasis Rfid Dan Aplikasi Mobile

Design And Implementation Of A Supabase-Based Backend For An Rfid-Based Smart Shopping Cart System With Mobile Application

Ni Made Sita Manika Putri
Prodi S1 Teknik Telekomunikasi
Fakultas Teknik Elektro
Universitas Telkom
sitamanikaa@student.telkomuniversity.ac.id

Ledya Novamizanti
Prodi S1 Teknik Telekomunikasi
Fakultas Teknik Elektro
Universitas Telkom
ledyaldn@telkomuniversity.ac.id

Abstrak

Toko ritel pakaian membutuhkan *backend* yang mampu menghubungkan pembacaan RFID, basis data, dan aplikasi *mobile* secara konsisten. Penelitian ini membahas perancangan *backend* menggunakan Supabase pada prototipe *smart shopping cart* berbasis RFID dan *Internet of Things* (IoT). *Backend* dirancang menggunakan PostgreSQL dengan tabel *products*, *rfid_tags*, *transactions*, *transaction_items*, *transaction_items_view*, dan *cart_events*. Alur transaksi menggunakan *cart_id* dan *transaction_id* untuk mendukung konsep *no-login*, sehingga transaksi berjalan tanpa menyimpan identitas pribadi pengguna. Pengujian dilakukan melalui verifikasi integritas data, privasi pengguna, *Row Level Security* (RLS), *Round Trip Time* (RTT), dan sinkronisasi status pembayaran. Hasil pengujian menunjukkan 10 item transaksi tersimpan konsisten dengan total Rp2.750.000, Delta_total = 0, dan status transaksi PAID. RTT rata-rata berada pada rentang 1.366 - 1.567 ms, sedangkan *delay* sinkronisasi pembayaran berada pada rentang 1,14 - 1,62 detik. Hasil ini menunjukkan bahwa Supabase mampu mendukung *backend cloud* yang *real-time*, menjaga integritas data, dan menerapkan minimisasi data pada prototipe *smart cart*.

Kata kunci : Supabase, PostgreSQL, RFID, *smart shopping cart*, *no-login*, RLS

Abstract

Clothing retail stores require a backend that consistently connects RFID readings, databases, and mobile applications. This study designs a Supabase-based backend for an RFID and Internet of Things (IoT) smart shopping cart prototype. PostgreSQL stores data in the products, rfid_tags, transactions, transaction_items, transaction_items_view, and cart_events tables. The transaction flow uses cart_id and transaction_id to support a no-login concept without storing users' personal identities. Evaluation covers data integrity, user privacy, Row Level Security (RLS), Round Trip Time (RTT), and payment synchronization. The results show that 10 transaction items were stored consistently with a total value of Rp2,750,000, Delta_total = 0, and PAID status. The average RTT ranged from 1,366 to 1,567 ms, while payment synchronization required 1.14 to 1.62 seconds. These results indicate that Supabase can support a real-time cloud backend, maintain transaction consistency, protect user privacy, and apply data minimization in the smart cart prototype.

***Keywords* : Supabase, PostgreSQL, RFID, *smart shopping cart*, *no-login*, RLS**

1. Pendahuluan

Sistem *smart shopping cart* berbasis IoT membutuhkan *backend* yang mampu menghubungkan data hasil pembacaan RFID dari perangkat ESP32 dengan aplikasi *mobile* secara konsisten dan *real-time*. Konsep integrasi IoT pada ritel mendukung proses transaksi yang lebih cepat

dan responsif karena data dari perangkat dapat diproses dan ditampilkan melalui aplikasi pengguna [1], [6]. Pada sistem berbasis RFID, data hasil pembacaan tag juga perlu dikelola secara hati-hati karena RFID mentah dapat mengandung pembacaan tidak relevan atau *stray read* [3], [4], [5].

Pada sistem ini, *backend* tidak hanya berfungsi sebagai tempat penyimpanan data, tetapi juga sebagai pusat pengelolaan transaksi, pencocokan data *tag* RFID dengan produk, pembaruan item belanja, validasi status pembayaran, serta penyediaan data yang dapat diakses oleh aplikasi *mobile*. Tanpa *backend* yang terstruktur, data hasil pembacaan RFID berisiko tidak tersimpan dengan benar, tidak sinkron dengan tampilan aplikasi, atau menyebabkan ketidaksesuaian antara produk yang terdeteksi dan total belanja pengguna.

Penelitian ini berfokus pada perancangan dan implementasi *backend* mengunakan Supabase untuk mendukung sistem keranjang belanja pintar pada ritel pakaian. Supabase digunakan karena menyediakan layanan basis data PostgreSQL, REST API, *Realtime*, serta pengaturan keamanan melalui *Row Level Security* (RLS). Struktur *backend* dirancang melalui beberapa tabel utama, yaitu *products*, *rfid_tags*, *transactions*, *transaction_items*, dan *cart_events*. Tabel-tabel tersebut saling berelasi untuk memastikan bahwa setiap *tag* RFID yang tervalidasi oleh ESP32 dapat dipetakan ke produk yang sesuai, dicatat dalam transaksi aktif, dan ditampilkan pada aplikasi *mobile* tanpa proses *login* pengguna.

Pendekatan *backend* pada penelitian ini juga menerapkan konsep *no-login session*, yaitu transaksi diidentifikasi berdasarkan sesi transaksi atau ID troli, bukan berdasarkan akun pribadi pelanggan. Pendekatan ini dipilih untuk mempercepat proses penggunaan sistem dan meminimalkan pengumpulan data pribadi. Selain itu, pengujian *backend* dilakukan untuk memverifikasi integritas data transaksi, sinkronisasi data secara *real-time*, latensi komunikasi, dan pembatasan akses data melalui RLS. Dengan demikian, *backend* menggunakan Supabase diharapkan mampu mendukung sistem *smart shopping cart* yang responsif, aman, dan sesuai dengan kebutuhan transaksi otomatis pada ritel pakaian.

2. Dasar Teori/Material dan Metodologi/Perancangan

2.1 Supabase dan PostgreSQL

Supabase merupakan *backend-as-a-service* yang menyediakan basis data PostgreSQL, REST API, autentikasi, penyimpanan file, dan fitur *realtime*. Pada penelitian ini, Supabase digunakan sebagai pusat data karena dapat menghubungkan ESP32 dan aplikasi *mobile* tanpa perlu membangun server *backend* manual. PostgreSQL dipilih karena mendukung skema relasional, *foreign key*, *view*, serta pengelolaan transaksi yang sesuai untuk pencatatan data produk dan transaksi belanja [8].

2.2 REST API dan *Realtime Synchronization*

REST API digunakan sebagai jalur komunikasi dari ESP32 ke Supabase. Setelah *tag* RFID dinyatakan valid oleh perangkat *edge*, ESP32 mengirimkan kode produk, ID transaksi, *quantity*, dan subtotal ke tabel transaksi. Pada sisi aplikasi *mobile*, fitur *realtime subscription* digunakan untuk menerima perubahan data secara langsung. Mekanisme ini membuat daftar belanja, total harga, dan status transaksi dapat diperbarui tanpa pengguna melakukan *refresh* manual [7].

2.3 *Row Level Security* dan *No-Login Session*

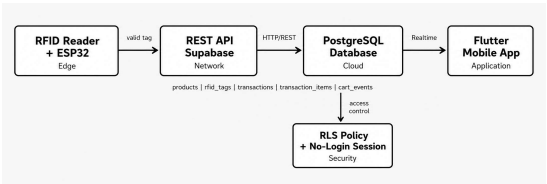
Row Level Security (RLS) merupakan mekanisme keamanan PostgreSQL yang membatasi akses data berdasarkan *policy* tertentu. Pada prototipe ini, RLS digunakan untuk membatasi operasi terhadap tabel *products*, *cart_events*, dan *transaction_items*. Selain itu, sistem menerapkan *no-login session*, yaitu transaksi tidak dikaitkan dengan akun pelanggan, melainkan dengan *cart_id* dan *transaction_id*. Pendekatan ini mengurangi data pribadi yang perlu disimpan dan mendukung konsep transaksi cepat pada ritel pakaian [5], [10].

Selain itu, sistem menerapkan *no-login session*, yaitu transaksi tidak dikaitkan dengan akun pelanggan, melainkan dengan *cart_id* dan *transaction_id*. Pendekatan ini memungkinkan pengguna melakukan proses belanja tanpa registrasi akun, sehingga alur transaksi menjadi lebih cepat dan data pribadi yang disimpan dapat dikurangi. Data yang tersimpan hanya berupa data operasional, seperti daftar produk, jumlah item, total harga, waktu transaksi, dan status pembayaran.

3. Perancangan *Backend* Menggunakan Supabase

3.1 Arsitektur *Backend*

Arsitektur *backend* dirancang untuk menerima data *tag* yang telah lolos validasi dari ESP32, menyimpannya pada PostgreSQL *database*, dan mengirimkan pembaruan data ke aplikasi Flutter secara *realtime*. Pada alur ini, ESP32 bertindak sebagai *edge device*, Supabase REST API sebagai jalur komunikasi jaringan, PostgreSQL sebagai *cloud database*, dan RLS sebagai lapisan pembatasan akses. Arsitektur *backend* ditunjukkan pada Gambar 1.



GAMBAR 1
Alur *Backend* Supabase Pada Sistem *Smart Shopping Cart*

Gambar 1 menunjukkan bahwa data *tag* valid dikirim dari RFID *Reader* dan ESP32 ke Supabase melalui REST API. Supabase menyimpan data pada tabel *products*, *rfid_tags*, *transactions*, *transaction_items*, dan *cart_events*. Aplikasi *mobile* kemudian menerima data transaksi secara *realtime*. Pembatasan akses dilakukan melalui *policy* RLS dan alur *no-login session* agar transaksi dapat berjalan tanpa identitas pribadi pengguna.

3.2 Rancangan Skema *Database*

Skema *database* dirancang secara relasional agar data produk, *tag* RFID, transaksi, detail transaksi, nota digital, dan aktivitas keranjang dapat dipisahkan tetapi tetap saling terhubung. Pemisahan tabel membantu menjaga modularitas, mengurangi duplikasi data, dan memudahkan proses verifikasi total transaksi. Entitas utama *backend* ditunjukkan pada Tabel 1.

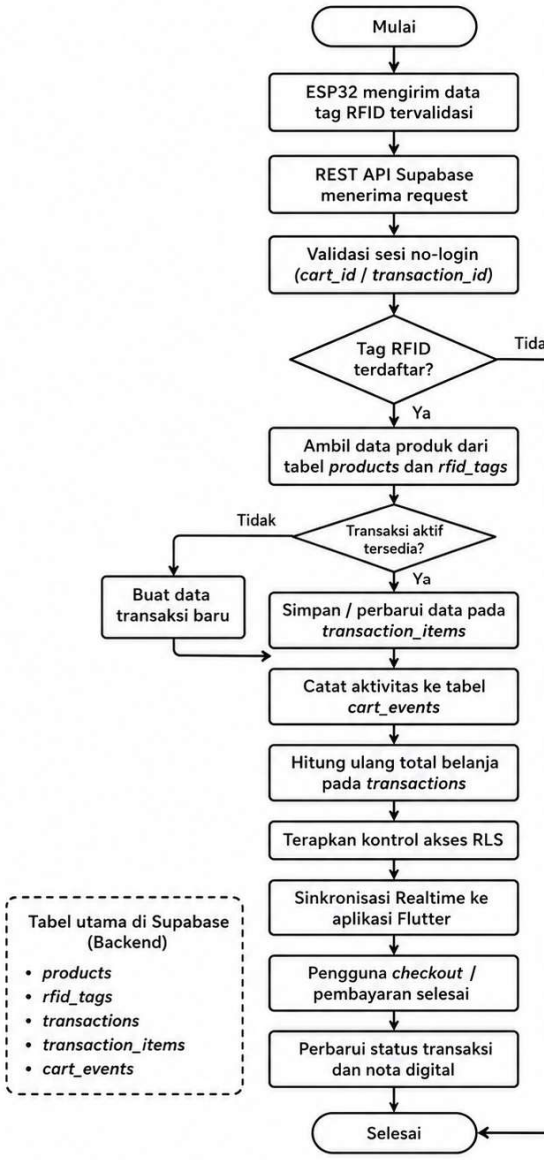
TABEL 1
Rancangan Entitas Utama *Backend* Supabase

Entitas	Fungsi	Data Kunci
products	Menyimpan data master produk yang ditampilkan pada aplikasi.	kode produk, nama, harga, stok, *image_url*
rfid_tags	Memetakan UID/EPC RFID ke produk tertentu.	*tag_uid*, *product_id*
transactions	Menyimpan sesi transaksi aktif dan status pembayaran.	*transaction_id*, *cart_id*, *total_amount*, status
transaction_items	Menyimpan rincian produk dalam satu transaksi.	*transaction_id*, *product_kode*, *quantity*, subtotal
transaction_items_view	Menggabungkan detail transaksi dan data produk untuk tampilan serta verifikasi.	nama produk, harga, *quantity*, subtotal
cart_events	Mencatat aktivitas keranjang untuk pelacakan pembacaan RFID dan pembatalan item.	*transaction_id*, epc, *event_type*, *created_at*
receipts_storage	Menyimpan nota digital berbasis transaksi.	*receipt_file*, *transaction_reference*

3.3 Alur Transaksi Berbasis *No-Login*

Alur transaksi *backend* dimulai saat ESP32 mengecek transaksi aktif untuk *cart_id* tertentu. Jika terdapat transaksi berstatus UNPAID, sistem menggunakan transaksi tersebut. Jika tidak ada, sistem membuat transaksi baru. Setelah *tag* RFID dinyatakan valid pada sisi ESP32, data dikirim ke *transaction_items* dan *event* dicatat pada *cart_events*. Aplikasi *mobile* membaca perubahan tersebut secara *realtime*. Ketika pengguna menyelesaikan *checkout*, status transaksi berubah dari UNPAID menjadi PAID.

Selain itu, *backend* juga mencatat setiap perubahan aktivitas ke dalam tabel *cart_events* sebagai *data* log sistem. Pencatatan ini digunakan untuk menelusuri proses pembacaan *tag*, pembaruan item, serta perubahan status transaksi. Setelah data item diperbarui pada *transaction_items*, total belanja pada tabel *transactions* dihitung ulang dan disinkronkan secara *realtime* ke aplikasi Flutter. Dengan mekanisme ini, aplikasi dapat menampilkan daftar produk dan total harga secara langsung tanpa memerlukan proses login pengguna. Diagram alir proses *backend* ditunjukkan pada Gambar 2.



GAMBAR 2
Diagram Alir Proses *Backend* Supabase

3.4 Keamanan dan Privasi Data

Backend Supabase menerapkan *Row Level Security* (RLS) untuk membatasi akses terhadap data transaksi, terutama pada tabel *transactions*, *transaction_items*, dan *cart_events*. Mekanisme ini penting karena Supabase menyediakan akses data melalui REST API yang terhubung langsung dengan PostgreSQL, sehingga setiap operasi baca dan tulis perlu dikontrol melalui kebijakan akses yang sesuai [2], [9].

Selain itu, sistem menggunakan konsep *no-login session* agar transaksi dapat berjalan tanpa menyimpan identitas pribadi pelanggan. Data yang disimpan hanya berupa data operasional, seperti *cart_id*, *transaction_id*, daftar produk, total harga, status pembayaran, dan nota digital, sehingga rancangan ini sesuai dengan prinsip minimisasi data [11].

4. Implementasi *Backend*

4.1 Implementasi Tabel dan Relasi

Implementasi *backend* dilakukan dengan membuat tabel utama pada Supabase PostgreSQL sesuai rancangan skema. Tabel *products* berfungsi sebagai data master produk, sedangkan *rfid_tags* menjadi jembatan antara UID/EPC RFID dan data produk. Tabel *transactions* menyimpan sesi belanja, sedangkan *transaction_items* menyimpan rincian produk pada setiap transaksi. *View transaction_items_view* digunakan untuk memudahkan aplikasi *mobile* menampilkan nama produk, harga, *quantity*, subtotal, dan total transaksi tanpa melakukan *query* relasional yang kompleks dari sisi aplikasi.

4.2 Implementasi Komunikasi REST API dan *Realtime*

ESP32 mengirimkan data hasil validasi ke Supabase melalui REST API. Data yang dikirim bukan pembacaan RFID mentah, melainkan *tag* yang telah dikonfirmasi oleh algoritma *temporal filtering* pada perangkat *edge*. Pendekatan ini mengurangi kemungkinan data *noise* masuk ke *database*. Pada sisi aplikasi *mobile*, *realtime subscription* digunakan untuk memantau perubahan tabel transaksi dan detail transaksi. Dengan cara ini, pengguna dapat melihat daftar barang dan total belanja secara langsung setelah *backend* menerima data baru.

4.3 Implementasi RLS dan Pembatasan Akses

Pembatasan akses dilakukan menggunakan RLS *policy*. *Policy* difokuskan pada kebutuhan prototipe, yaitu membaca data produk, menambahkan atau membaca aktivitas keranjang, serta menambahkan, membaca, atau menghapus item transaksi sesuai sesi. Ringkasan RLS ditunjukkan pada Tabel 2.

TABEL 2
Ringkasan *Policy* RLS Pada *Backend* Supabase

Tabel	Policy	Fungsi
products	SELECT	Mengizinkan aplikasi membaca data produk.
cart_events	INSERT dan SELECT	Mengizinkan sistem menambahkan dan membaca aktivitas keranjang.
transaction_items	SELECT, INSERT, DELETE	Mengizinkan pembacaan, penambahan, dan pembatalan item transaksi sesuai sesi.

5. Pengujian dan Analisis

5.1 Verifikasi Integritas Data Transaksi

Verifikasi integritas data dilakukan dengan membandingkan data pada *transaction_items_view*, *tabel transactions*, dan tampilan aplikasi *mobile*. Parameter yang diperiksa meliputi jumlah item, *quantity*, kode produk, subtotal, total transaksi, dan status pembayaran. Ringkasan hasil verifikasi ditunjukkan pada Tabel 3.

TABEL 3
Hasil Verifikasi Integritas Data Transaksi

No.	Parameter	Hasil Verifikasi	Keterangan
1	Jumlah item transaksi	10 item	Sesuai
2	*Quantity* setiap item	1	Sesuai
3	Total *transaction_items_view*	Rp2.750.000	Sesuai
4	Total pada tabel *transactions*	Rp2.750.000	Sesuai
5	Delta_total	0	Sesuai
6	Status transaksi	PAID	Sesuai
7	Data pribadi pengguna	Tidak tersimpan pada alur utama	Sesuai

Hasil pada Tabel 3 menunjukkan bahwa *backend* mampu menyimpan transaksi secara konsisten. Seluruh item berada pada satu *transaction_id* yang sama, total detail transaksi sama dengan total pada tabel induk, dan nilai Delta_total = 0. Hal ini menunjukkan bahwa relasi antara *transactions* dan *transaction_items* berjalan sesuai rancangan.

5.2 Verifikasi Privasi dan Minimisasi Data

Verifikasi privasi dilakukan dengan memeriksa struktur data pada alur utama transaksi. Data yang tersimpan hanya berupa data operasional, yaitu ID transaksi, *cart_id*, daftar produk, *quantity*, subtotal, status pembayaran, dan informasi nota. Data seperti nama, email, nomor telepon, dan alamat tidak menjadi syarat penggunaan sistem. Hasil verifikasi ditunjukkan pada Tabel 4.

TABEL 4
Hasil Verifikasi Privasi Pada Alur Transaksi Utama

Komponen	Data yang Tersimpan	Status
transactions	ID transaksi, tanggal, *cart_id*, status, total transaksi, kode transaksi, *receipt_snapshot*	Memenuhi
transaction_items	ID transaksi, kode produk, jumlah, subtotal, dan waktu pencatatan	Memenuhi
cart_events	ID transaksi, EPC, jenis event, dan waktu event	Memenuhi
transaction_items_view	Gabungan data produk dan item transaksi untuk tampilan aplikasi	Memenuhi
Identitas pengguna	Tidak menyimpan nama, email, nomor telepon, atau alamat pada alur utama	Memenuhi
Akses nota digital	Berbasis ID transaksi, bukan akun pengguna	Memenuhi

5.3 Pengujian Latensi dan Sinkronisasi Status Pembayaran

Pengujian latensi dilakukan untuk memastikan komunikasi antara perangkat *edge*, Supabase, dan aplikasi *mobile* tetap berada dalam batas toleransi sistem. Hasil pengujian menunjukkan RTT rata-rata berada pada rentang 1.366 - 1.567 ms. Nilai tersebut masih berada di bawah batas spesifikasi 4.000 ms. Selain itu, sinkronisasi status pembayaran dari UNPAID menjadi PAID membutuhkan waktu 1,14 - 1,62 detik. Ringkasan hasil pengujian ditunjukkan pada Tabel 5.

TABEL 5
Ringkasan Hasil Pengujian Komunikasi *Backend*

Parameter	Hasil	Batas Spesifikasi	Status
RTT rata-rata seluruh skenario	1.366 - 1.567 ms	<= 4.000 ms	Tercapai
Jumlah pengukuran RTT	38 pengukuran	Seluruhnya valid	Tercapai
Sinkronisasi status pembayaran	1,14 - 1,62 detik	<= 4 detik	Tercapai
Integritas total transaksi	Delta_total = 0	Data konsisten	Tercapai

Hasil tersebut menunjukkan bahwa Supabase mampu mendukung pertukaran data prototipe secara cukup responsif. Walaupun demikian, konfigurasi akses *backend* masih perlu diperketat untuk penggunaan produksi, terutama pada pembatasan API *key*, validasi *server-side*, *audit policy* RLS, dan pemisahan akses antara perangkat *edge*, aplikasi pengguna, serta admin toko.

6. Kesimpulan

Backend menggunakan Supabase berhasil dirancang dan diimplementasikan sebagai pusat penyimpanan serta sinkronisasi data pada sistem *smart shopping cart* berbasis RFID dan aplikasi *mobile*. Skema *database* relasional dapat menghubungkan data produk, RFID, transaksi,

detail transaksi, aktivitas keranjang, dan nota digital secara terstruktur. Penggunaan *cart_id* dan *transaction_id* mendukung alur transaksi *no-login* sehingga sistem dapat berjalan tanpa menyimpan identitas pribadi pengguna pada alur utama. Hasil pengujian menunjukkan bahwa integritas data transaksi telah terpenuhi. Sebanyak 10 item tersimpan pada satu sesi transaksi dengan total Rp2.750.000, status PAID, dan Delta_total = 0. *Backend* juga memenuhi target komunikasi karena RTT rata-rata berada pada rentang 1.366 - 1.567 ms dan sinkronisasi status pembayaran berada pada rentang 1,14 - 1,62 detik, atau masih di bawah batas 4 detik. Dengan demikian, Supabase layak digunakan sebagai *backend cloud* pada prototipe *smart cart* karena mampu menyediakan *database* relasional, REST API, *real-time synchronization*, dan RLS. Pengembangan lanjutan perlu difokuskan pada penguatan keamanan produksi, seperti penggunaan *service role* hanya pada server, validasi *request* yang lebih ketat, *audit policy* RLS, dan integrasi dengan sistem inventori serta pembayaran komersial.

REFERENSI

[1] Alfian, G., Syafrudin, M., Yoon, B., dan Rhee, J. 2019. False Positive RFID Detection Using Classification Models. *Applied Sciences (Switzerland)*. **9:6 1154**. doi: 10.3390/app9061154.

[2] Martinus, Wahab, M. S., Yudi, dan Ham, H. 2021. Data Transmission Using RFID System on Smart Shopping Carts for Checkout Process Efficiency in Supermarket at Indonesia. *Procedia Computer Science*. **179:902-912**. doi: 10.1016/j.procs.2021.01.080.

[3] Priyambodo, A., Usman, K., dan Novamizanti, L. 2020. Implementation of Android-Based QR Code in the Presence System. *Jurnal Teknologi Informasi dan Ilmu Komputer*. **7:5 1011-1020**. doi: 10.25126/jtiik.202072337.

[4] Republik Indonesia. 2019. *Peraturan Pemerintah Republik Indonesia Nomor 71 Tahun 2019 tentang Penyelenggaraan Sistem dan Transaksi Elektronik*. Lembaran Negara Republik Indonesia Tahun 2019 Nomor 185. [Online] Available at: https://peraturan.bpk.go.id/Details/122030/pp-no-71-tahun-2019 [Accessed 7 May 2026].

[5] Republik Indonesia. 2022. *Undang-Undang Republik Indonesia Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi*. Lembaran Negara Republik Indonesia Tahun 2022 Nomor 196. [Online] Available at: https://peraturan.bpk.go.id/Details/229798/uu-no-27-tahun-2022 [Accessed 7 May 2026].

[6] Shahroz, M., Mushtaq, M. F., Ahmad, M., Ullah, S., Mehmood, A., dan Choi, G. S. 2020. IoT-Based Smart Shopping Cart Using Radio Frequency Identification. *IEEE Access*. **8:68426-68438**. doi: 10.1109/ACCESS.2020.2986681.

[7] Supabase. 2026. *Data REST API*. [Online] Available at: https://supabase.com/docs/guides/api [Accessed 18 May 2026].

[8] Supabase. 2026. *Database | Supabase Docs*. [Online] Available at: https://supabase.com/docs/guides/database/overview [Accessed 8 June 2026].

[9] Supabase. 2026. *Postgres Changes | Supabase Realtime Docs*. [Online] Available at: https://supabase.com/docs/guides/realtime/postgres-changes [Accessed 8 June 2026].

[10] Supabase. 2026. *Row Level Security | Supabase Docs*. [Online] Available at: https://supabase.com/docs/guides/database/postgres/row-level-security [Accessed 8 June 2026].

[11] Zulfiqar, M. I., Khalid, A., Siddig, A., Nawaz, M. J., dan Saay, S. 2025. AI-Driven Smart Shopping Carts With Real-Time Tracking and Inventory Forecasting for Enhanced Retail Efficiency. *IEEE Access*. **13:55576-55585**. doi: 10.1109/ACCESS.2025.3553854.