

Perancangan dan Pengembangan Aplikasi Reservasi Restoran X pada Platform iOS berbasis App Clip

Design and Development of the Restaurant X Reservation Application on the iOS Platform using App Clip

¹Megan Rochella*, ²Kartika Gianina Tileng

^{1,2}Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Ciputra Surabaya

^{1,2}CitraLand CBD Boulevard, Surabaya, Jawa Timur, 60219, Indonesia

*e-mail: meganrochella@gmail.com

(received: 16 December 2025, revised: 20 January 2026, accepted: 25 January 2026)

Abstrak

Industri makanan dan minuman di Indonesia menunjukkan pertumbuhan yang signifikan dengan 4,85 juta unit usaha pada tahun 2023, namun banyak restoran masih mengandalkan sistem reservasi manual yang menghambat efisiensi operasional. Meskipun penetrasi internet mencapai 79,5% dengan *smartphone* sebagai perangkat akses utama (83,39%), adopsi aplikasi *mobile* menghadapi hambatan karena friksi dalam proses instalasi. Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi reservasi restoran berbasis iOS menggunakan teknologi App Clip yang terintegrasi dengan *dashboard* admin secara *real-time*. Pengembangan sistem menggunakan pendekatan arsitektur MVVM, dibangun menggunakan Swift dengan SwiftUI untuk antarmuka, Golang untuk *backend*, dan PostgreSQL untuk basis data. Sistem mencakup aplikasi pemesanan dari pelanggan, selanjutnya aplikasi manajemen reservasi di pihak restoran, pembayaran uang muka melalui QRIS yang ditampilkan melalui aplikasi, pemilihan meja berdasarkan preferensi *ambience* dari pelanggan, dan mekanisme pencegahan *overbooking* otomatis, serta finalisasi pemesanan saat waktu sudah selesai. Evaluasi pengembangan sendiri dilakukan melalui pengujian *usability* berbasis tugas dengan 7 responden (4 admin, 3 pelanggan). Hasil menunjukkan tingkat penyelesaian tugas 100% pada kedua antarmuka, melebihi rata-rata *benchmark* 78%, dengan akses App Clip yang berhasil menunjukkan efektivitasnya sebagai metode akses cepat tanpa instalasi. Penelitian ini berkontribusi pada dokumentasi implementasi App Clip dalam sistem reservasi *mobile* dan solusi manajemen reservasi terintegrasi yang dapat diadaptasi ke sektor *hospitality* lainnya.

Kata kunci: app clip, iOS, MVVM, pengujian *usability*, reservasi restoran

Abstract

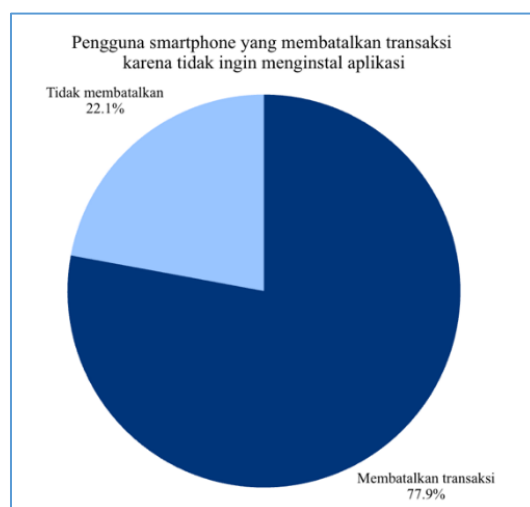
The food and beverage industry in Indonesia has shown significant growth, with 4.85 million business units in 2023. However, many restaurants still rely on manual reservation systems, which hinder operational efficiency. Despite internet penetration reaching 79.5%, with smartphones as the primary access device (83.39%), mobile application adoption faces barriers due to friction in the installation process. This study aims to design and implement an iOS-based restaurant reservation application using App Clips technology, integrated with a real-time admin dashboard. The system was developed using the MVVM architecture, with Swift and SwiftUI for the user interface, Golang for the backend, and PostgreSQL for the database. The system includes a customer-facing reservation app, a restaurant-side reservation management app, advance payment via QRIS displayed through the app, table selection based on customer ambience preferences, an automatic overbooking prevention mechanism, and finalization of reservations once the allotted time is complete. Development evaluation was conducted using task-based usability testing with seven respondents (four admins and three customers). The results showed a 100% task completion rate on both interfaces, exceeding the benchmark average of 78%, while App Clip access successfully demonstrated its effectiveness as a quick-access method without installation. This study contributes to the documentation of App Clip implementation in mobile reservation systems and presents an integrated reservation management solution that can be adapted to other sectors within the hospitality industry.

Keywords: app clip, iOS, MVVM, reservasi restoran, usability testing

1 Pendahuluan

Industri penyediaan makanan dan minuman di Indonesia menunjukkan pertumbuhan signifikan dengan 4,85 juta unit usaha pada tahun 2023, termasuk 1,2 juta restoran dan rumah makan [1]. Pertumbuhan ini menciptakan persaingan ketat yang mendorong kebutuhan peningkatan efisiensi operasional, salah satunya melalui pengelolaan reservasi yang strategis [2]. Sistem reservasi digital dapat meningkatkan kepuasan pelanggan dengan mengurangi waktu tunggu, memberikan kepastian ketersediaan meja, serta membantu perencanaan operasional restoran [3], [4].

Namun, banyak restoran di Indonesia masih menggunakan sistem manual melalui telepon atau kunjungan langsung, yang menimbulkan masalah seperti pencatatan tidak terorganisir, pelayanan lambat, dan risiko kehilangan pelanggan [5]. Padahal, potensi adopsi solusi *mobile* sangat besar. Di Indonesia, penetrasi internet mencapai 79,5% dengan 221,5 juta pengguna [6]. *Smartphone* mendominasi akses internet sebesar 83,39% [7]. Selain itu, pengguna transaksi QRIS di Indonesia mencapai 51,43 juta per Juli 2024 [8]. Meskipun demikian, terdapat hambatan signifikan dalam adopsi aplikasi *mobile*. Palau-Saumell et al. [9] mengidentifikasi *effort expectancy* (persepsi kemudahan) sebagai faktor kritis yang mempengaruhi niat penggunaan. Hal ini diperkuat oleh penelitian Heady (2021) menunjukkan bahwa 77,9% pengguna *smartphone* membatalkan transaksi karena hambatan instalasi aplikasi [10], seperti ditunjukkan pada Gambar 1. Setiap langkah tambahan seperti *download* dan instalasi aplikasi menjadi *friction point* yang menghambat transaksi spontan atau *one-time* [11].



Gambar 1 Dampak hambatan instalasi aplikasi terhadap transaksi *mobile*

Teknologi App Clip dirancang untuk mengatasi hambatan ini dengan menyediakan akses instan tanpa instalasi aplikasi penuh [12]. Meskipun teknologi ini telah tersedia sejak iOS 14 (2020) [12], adopsinya dalam aplikasi reservasi restoran di Indonesia masih minimal. Terdapat *gap* penelitian dalam tiga aspek: (1) implementasi App Clip dengan integrasi *backend real-time* untuk sistem reservasi restoran belum terdokumentasi dengan baik, (2) panduan arsitektur MVVM untuk *dual-interface* (App Clip dan *full app*) masih terbatas, dan (3) evaluasi efektivitas App Clip dalam aplikasi *hospitality* berbasis *usability testing* belum banyak dilakukan. Berdasarkan *gap* tersebut, penelitian ini merumuskan tiga *research questions* (RQ): (RQ1) Bagaimana merancang arsitektur sistem reservasi restoran berbasis iOS dengan App Clip menggunakan pendekatan MVVM yang terintegrasi dengan *dashboard admin real-time*? (RQ2) Bagaimana mengimplementasikan fitur-fitur utama sistem reservasi dalam aplikasi iOS? (RQ3) Bagaimana tingkat *usability* aplikasi yang dikembangkan berdasarkan *task-based usability testing* pada antarmuka pelanggan dan admin?

Penelitian ini bertujuan merancang dan mengimplementasikan aplikasi reservasi restoran berbasis iOS dengan App Clip yang terintegrasi dengan *dashboard admin real-time*, memungkinkan reservasi tanpa instalasi dan pengelolaan ketersediaan meja serta konfirmasi pembayaran *down payment*. Penelitian ini memberikan tiga kontribusi utama: (1) teoretis, yaitu dokumentasi

implementasi App Clip dengan arsitektur MVVM *dual-interface* untuk sistem reservasi *real-time*, mengisi *gap* literatur teknologi *instant app* dalam aplikasi *hospitality*; (2) metodologis, yaitu demonstrasi bahwa *task-based usability testing* dengan 7 responden mampu menghasilkan wawasan mendalam yang ditunjukkan oleh *task completion rate* 100% serta teridentifikasinya tiga area perbaikan, memberikan *framework* evaluasi yang efisien untuk aplikasi serupa; (3) praktis, yaitu sistem reservasi berbasis App Clip dan *dashboard* admin *real-time* yang dapat diadaptasi sektor *hospitality* lainnya seperti hotel dan spa.

2 Tinjauan Literatur

Penelitian ini menggunakan pendekatan *Challenge Based Learning* (CBL) yang dikembangkan oleh Apple Inc. CBL merupakan pendekatan yang melibatkan pembelajar dalam mengidentifikasi tantangan nyata, melakukan investigasi kolaboratif, dan mengembangkan solusi inovatif yang relevan dengan konteks dunia nyata [13], [14]. *Framework* CBL terdiri dari tiga fase utama, yaitu *Engage* untuk mengidentifikasi *big idea* dan penetapan *challenge*, *Investigate* untuk pengembangan *guiding questions* dan pengumpulan *resources*, serta *Act* untuk pengembangan dan evaluasi solusi, dengan evaluasi terintegrasi di setiap tahapan [15]. Implementasi CBL dalam penelitian ini dimulai dari identifikasi permasalahan efisiensi pelayanan restoran sebagai *big idea* hingga pengembangan sistem reservasi berbasis iOS dengan evaluasi melalui *usability testing*.

Arsitektur aplikasi yang dikembangkan menggunakan pola *Model-View-ViewModel* (MVVM) yang memfasilitasi pemisahan pengembangan antarmuka pengguna dari logika bisnis [15]. Arsitektur ini terdiri dari tiga komponen utama yaitu *Model* sebagai lapisan data dan logika bisnis, *ViewModel* sebagai mediator yang menangani *presentation logic* dan menjembatani *Model* dengan *View*, serta *View* untuk menampilkan antarmuka pengguna berdasarkan data dari *ViewModel*. MVVM bekerja efektif dengan pemrograman reaktif melalui mekanisme *data binding* yang memungkinkan perubahan *Model* mempengaruhi *View* secara otomatis [16].

Teknologi utama yang digunakan adalah App Clip, fitur iOS yang memungkinkan pengguna mengakses fungsionalitas spesifik aplikasi secara instan tanpa perlu mengunduh aplikasi penuh [12]. Teknologi ini diperkenalkan oleh Apple pada WWDC 2020 sebagai bagian dari iOS 14 dengan konsep "*lightweight version of your app that offers users some of its functionality when and where they need it*" [12]. App Clip dirancang untuk menyediakan pengalaman *just-in-time* yang memungkinkan pengguna menyelesaikan tugas tertentu dalam hitungan detik tanpa hambatan instalasi aplikasi konvensional. Apple menerapkan beberapa pembatasan pada App Clip, dimana *Limit App Tracking* selalu diaktifkan sehingga IDFA (*Identifier for Advertisers*) dan IDFV (*Identifier for Vendors*) mengembalikan nilai kosong, mencegah pelacakan pengguna lintas aplikasi [12]. Selain itu, App Clip juga tidak dapat melakukan aktivitas latar belakang dan tidak disertakan dalam *backup* iOS.

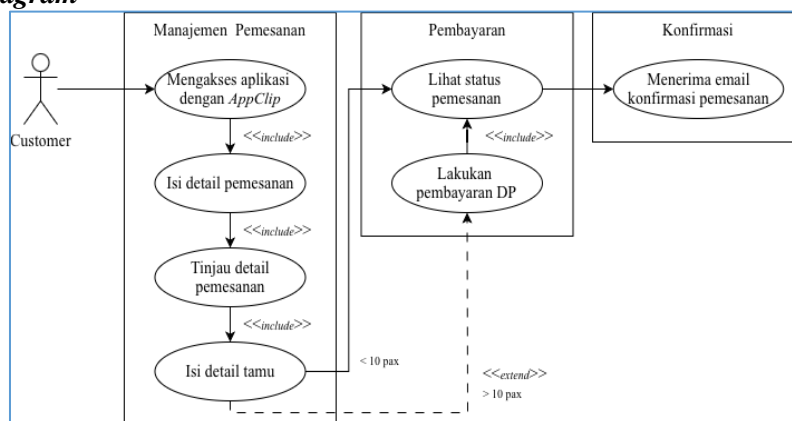
Teknologi pengembangan perangkat lunak mencakup bahasa pemrograman, *framework*, lingkungan pengembangan, serta layanan pihak ketiga yang digunakan untuk membangun sistem [17]. Dalam penelitian ini, pemilihan teknologi disesuaikan dengan kebutuhan pengembangan aplikasi reservasi restoran berbasis App Clip. *Mobile development* menggunakan Swift dan SwiftUI sebagai teknologi *native* iOS yang memberikan performa optimal dan akses penuh ke fitur App Clip [18]. *Backend development* menggunakan Golang 1.22 dengan *Gin Framework* karena kemampuan *concurrency* melalui *goroutines* yang efisien untuk menangani banyak permintaan secara bersamaan [19], [20]. PostgreSQL 16 dipilih sebagai *database* karena dukungan *ACID compliance* dan *complex queries* untuk validasi ketersediaan meja secara *real-time* [21]. *Authentication* menggunakan JWT untuk pengelolaan sesi secara *stateless* tanpa memerlukan penyimpanan *session* di sisi server. Layanan pihak ketiga mencakup QRIS untuk pembayaran, APNs untuk notifikasi *real-time*, SMTP untuk *email*, dan WhatsApp Business API untuk *follow-up* reservasi. Pengujian dilakukan melalui TestFlight dengan *version control* menggunakan Git sebagai standar industri untuk manajemen *source code*.

3 Metode Penelitian

A. Analisis Kebutuhan Sistem

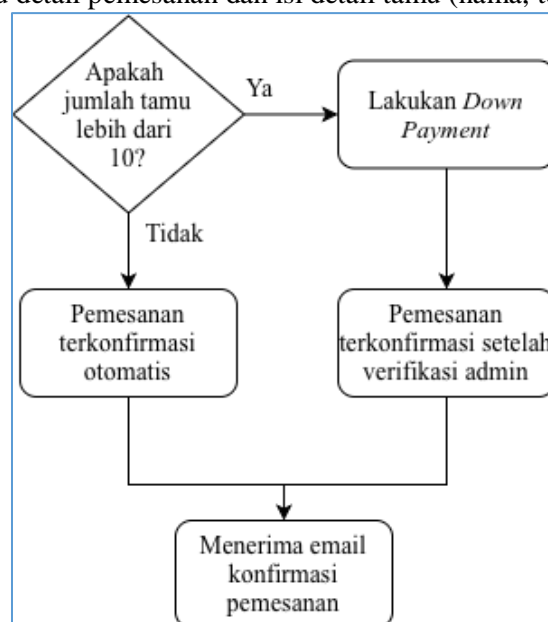
Analisis kebutuhan sistem dilakukan melalui survei langsung di lokasi restoran dengan melibatkan 3 *stakeholder* kunci yang terdiri dari manajer restoran, staf *finance*, dan *greeter*. Teknik survei dipilih untuk mengidentifikasi kebutuhan dan keinginan *stakeholder* secara sistematis, yang merupakan aktivitas fundamental dalam proses analisis kebutuhan [22]. Kebutuhan fungsional meliputi: (1) akses pemesanan melalui App Clip tanpa instalasi aplikasi penuh, (2) validasi ketersediaan meja secara *real-time* untuk mencegah *overbooking*, (3) pembayaran *down payment* via QRIS untuk tamu lebih dari 10 orang dengan *countdown timer* 5 menit, (4) verifikasi pembayaran manual oleh admin dengan perubahan status (*Unconfirmed DP/Confirmed DP*), (5) komunikasi WhatsApp untuk *follow-up* pembayaran, (6) penempatan meja berdasarkan preferensi *ambience* pelanggan, (7) pengelolaan meja *walk-in* dan reservasi dengan *auto-clear* setelah 2 jam, (8) notifikasi *push real-time* ke admin untuk pemesanan baru, (9) *dashboard* statistik dengan visualisasi, dan (10) ekspor data dalam format PDF.

B. Use Case Diagram



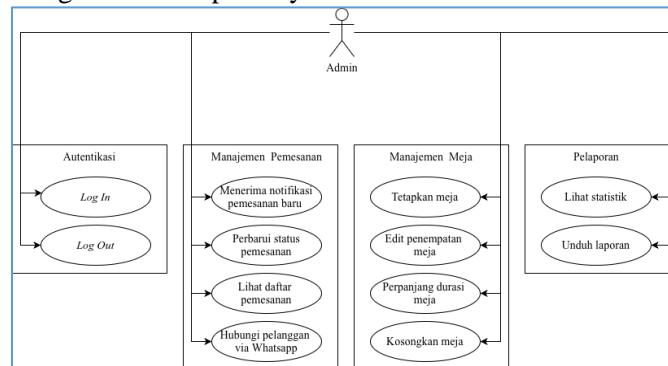
Gambar 2 Use case diagram antarmuka pelanggan

Use cases pelanggan pada kategori manajemen pemesanan dimulai dengan mengakses aplikasi melalui App Clip, diikuti isi detail pemesanan yang mencakup pilih tanggal & waktu, jumlah tamu, area tempat duduk, preferensi area merokok, acara spesial, dan permintaan *ambience*. Pelanggan kemudian melakukan tinjau detail pemesanan dan isi detail tamu (nama, telepon, *email*).



Gambar 3 Decision logic pembayaran down payment berdasarkan jumlah tamu

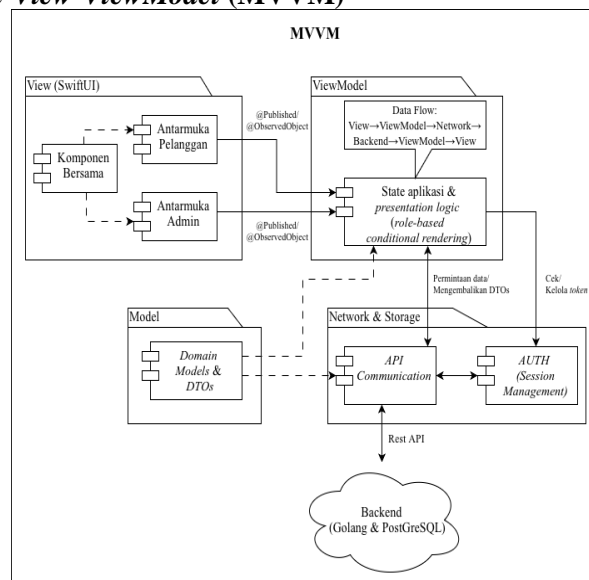
Kategori pembayaran berfungsi sebagai *conditional process* berdasarkan jumlah tamu seperti ditunjukkan pada Gambar 3. Sistem secara otomatis mengecek jumlah tamu sebagai *decision point*. Untuk reservasi lebih dari 10 orang, pelanggan diarahkan untuk melakukan pembayaran *down payment* yang mencakup lihat *barcode QRIS*, unduh *barcode*, dan *timer* pembayaran (5 menit), kemudian lihat status pemesanan untuk menunggu verifikasi admin. Untuk reservasi 10 orang atau kurang, pemesanan langsung terkonfirmasi otomatis tanpa pembayaran. Kategori konfirmasi mencakup pengiriman *email* konfirmasi otomatis dengan detail pemesanan untuk semua reservasi yang terintegrasi dari kedua alur. Untuk reservasi dengan *down payment*, *email* konfirmasi tambahan dikirim setelah admin mengubah status pembayaran.



Gambar 4 Use case diagram antarmuka admin

Use cases admin mencakup empat kategori utama. Pertama, autentikasi dan manajemen sesi meliputi *login* dan *logout*. Kedua, manajemen pemesanan dan pembayaran yang mencakup terima notifikasi pesan baru, perbarui status pemesanan (*Unconfirmed DP/Confirmed DP*), lihat daftar pemesanan, dan hubungi pelanggan via WhatsApp untuk tindak lanjut pembayaran. Ketiga, manajemen meja yang memungkinkan admin untuk menetapkan meja berdasarkan preferensi *ambience* atau untuk tamu *walk-in*, *edit* penempatan meja, perpanjang durasi meja, dan kosongkan meja. Keempat, pelaporan mencakup lihat statistik pemesanan dan unduh laporan PDF.

C. Perancangan Model-View-ViewModel (MVVM)



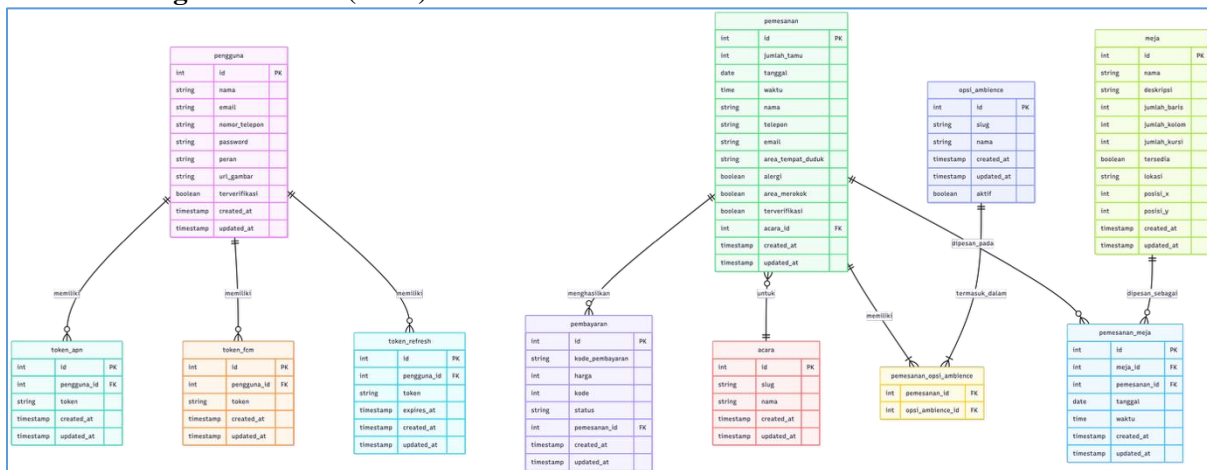
Gambar 5 Arsitektur MVVM

Lapisan *View* menggunakan *SwiftUI framework* untuk menampilkan antarmuka pengguna dengan implementasi terpisah untuk admin dan antarmuka pelanggan, namun menggunakan komponen bersama untuk konsistensi. *View* berkomunikasi dengan *ViewModel* melalui *data binding* menggunakan *@Published* dan *@ObservedObject* sehingga UI otomatis *ter-update* ketika data berubah. Lapisan *ViewModel* mengelola *state* aplikasi dan *presentation logic* dengan *role-based*

<http://sistemasi.ftik.unisi.ac.id>

conditional rendering untuk mendukung *dual-interface*. *ViewModel* bertanggung jawab mengambil data, mentransformasi untuk presentasi, menangani *business logic*, dan mengelola *loading states*. Lapisan *Model* terdiri dari *domain models* dan *DTOs* yang merepresentasikan *business entities* dan digunakan untuk komunikasi dengan API. Lapisan *Network & Storage* menangani komunikasi dengan *backend API* (Golang dan PostgreSQL) dan penyimpanan data lokal menggunakan *Authentication* untuk *user session management*. *Data flow* mengikuti alur *unidirectional*: (1) *User* melakukan interaksi di *View*, (2) *View* memanggil *ViewModel*, (3) *ViewModel* request data ke API *Communication*, (4) API *Communication* mengembalikan *response*, (5) *ViewModel* meng-update *state*, dan (6) *View* ter-update otomatis melalui *SwiftUI data binding*. Implementasi *MVVM* dengan *dual-interface* ini menghasilkan *code reusability* yang tinggi karena lapisan *Model* dan *ViewModel* dapat digunakan bersama oleh admin dan antarmuka pelanggan, sementara hanya lapisan *View* yang disesuaikan untuk setiap *user role*.

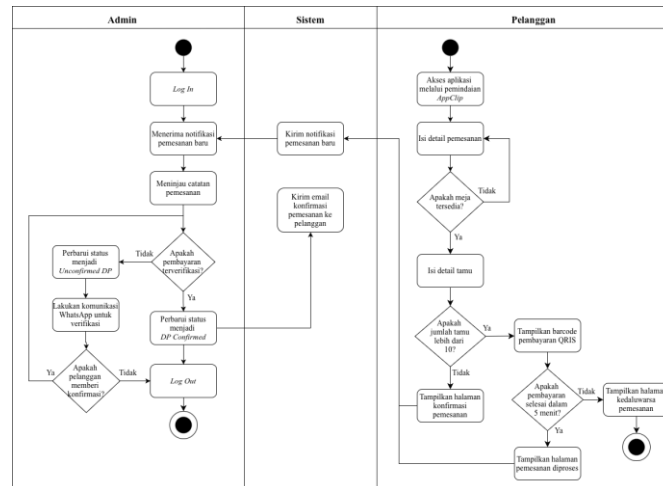
D. Perancangan Database (ERD)



Gambar 6 Entity relationship diagram (ERD)

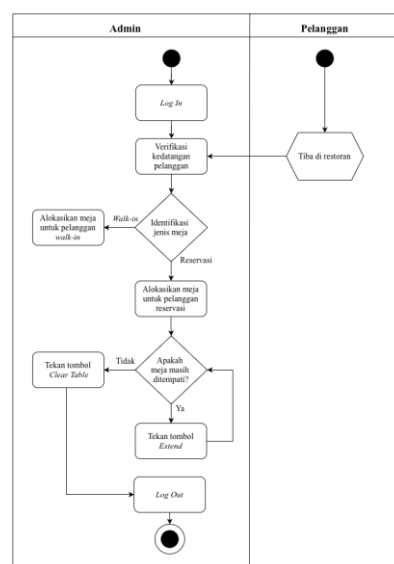
Database dibangun menggunakan 11 tabel seperti ditunjukkan pada Gambar 6, yang dikategorikan menjadi empat entitas utama yaitu pengguna, pemesanan, meja, pembayaran, dua tabel referensi yaitu acara dan opsi_ambience, dua *junction table* yaitu pemesanan_opsi_ambience dan pemesanan_meja, serta tiga tabel token untuk autentikasi dan notifikasi yaitu token_apn, token_fcm dan token_refresh. Tabel pemesanan merupakan entitas pusat yang menyimpan informasi reservasi dengan atribut jumlah_tamu, tanggal, waktu, detail kontak (nama, telepon, email), preferensi (area_tempat_duduk, alergi, area_merokok), status verifikasi (terverifikasi), dan referensi ke acara_id. Tabel ini berelasi *one-to-many* dengan tabel pembayaran yang menyimpan atribut kode_pembayaran, harga, kode, dan status untuk pelacakan transaksi. Tabel meja menyimpan atribut kapasitas (jumlah_baris, jumlah_kolom, jumlah_kursi), status ketersediaan (tersedia), lokasi, dan koordinat posisi (posisi_x, posisi_y). Relasi antara pemesanan dan meja dijumpai oleh *junction table* pemesanan_meja untuk mengelola alokasi meja pada tanggal dan waktu tertentu. Relasi *many-to-many* antara pemesanan dan opsi_ambience diimplementasikan melalui *junction table* pemesanan_opsi_ambience. Tabel pengguna dengan atribut peran untuk *role-based access control* berelasi *one-to-many* dengan tiga tabel token yang mendukung autentikasi (token_refresh) dan *push notifications* (token_apn untuk iOS, token_fcm untuk cross-platform). Setiap tabel dilengkapi *primary key* (id), *timestamps* (created_at, updated_at), serta *foreign keys* untuk menjaga integritas data.

E. Activity Diagram



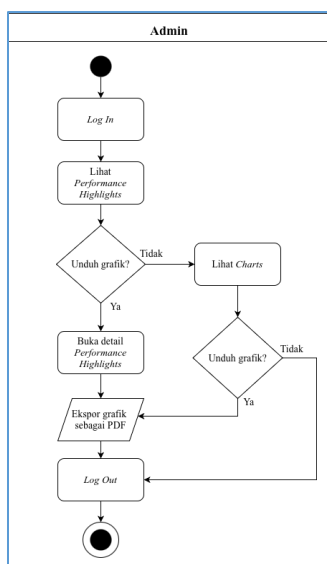
Gambar 7 Activity diagram manajemen pemesanan

Activity Diagram Manajemen Pemesanan menggambarkan alur kerja proses reservasi dengan tiga swimlanes yaitu Admin, Sistem, dan Pelanggan. Pada sisi pelanggan, proses dimulai dengan akses aplikasi melalui pemindaian App Clip, pengisian detail pemesanan, dan validasi ketersediaan meja oleh sistem. Jika meja tersedia, pelanggan mengisi detail tamu. Jika tidak, kembali ke input detail. Selanjutnya terdapat *decision point* jumlah tamu yaitu jika lebih dari 10 orang, sistem menampilkan *barcode* QRIS dengan *timer* 5 menit yang mengarah ke halaman kadaluarsa (jika *timeout*) atau halaman pemesanan diproses (jika selesai). Jika 10 atau kurang, sistem langsung menampilkan halaman konfirmasi. Pada sisi admin dalam Activity Diagram Manajemen Pemesanan, setelah *login* admin menerima notifikasi pesanan baru dan meninjau catatan pemesanan. *Decision point* verifikasi pembayaran menentukan status yaitu jika terverifikasi maka status menjadi *DP Confirmed* dan sistem mengirim *email* konfirmasi, jika tidak maka status menjadi *Unconfirmed DP*. Admin kemudian melakukan komunikasi WhatsApp untuk verifikasi kepada pelanggan sebelum *logout*.



Gambar 8 Activity diagram manajemen meja

Activity Diagram Manajemen Meja menggambarkan koordinasi penugasan meja dengan dua swimlanes yaitu Admin dan Pelanggan. Proses dimulai dari dua titik awal yaitu admin *login* dan pelanggan tiba. Admin memverifikasi kedatangan, kemudian sistem mengidentifikasi jenis meja melalui *decision point* *Walk-in* atau *Reservasi* untuk mengalokasikan meja sesuai kategori. Kedua alur bergabung pada *decision point* status meja yaitu jika tidak ditempati maka tekan *Clear Table*, jika masih ditempati maka tekan *Extend* untuk perpanjang durasi, dan diakhiri dengan *logout*.



Gambar 9 Activity diagram statistik

Activity Diagram Statistik menunjukkan alur admin dalam menganalisis kinerja bisnis melalui satu swimlane Admin. Setelah login, admin melihat Performance Highlights yang menampilkan ringkasan metrik utama dengan decision point untuk unduh grafik atau melihat charts. Jika memilih melihat charts, admin dapat melihat visualisasi data statistik dengan opsi unduh. Jika memilih unduh (baik dari highlights maupun charts), sistem membuka detail grafik dan melakukan ekspor sebagai PDF. Proses diakhiri dengan logout.

F. Perancangan App Clip



Gambar 10 Alur invokasi app clip

Proses dimulai dari invokasi menggunakan QR Code yang ditempatkan di meja dan pintu masuk restoran. Pemindaian kode memicu munculnya kartu App Clip yang meluncurkan aplikasi melalui URL dengan skema kustom (`appclips://restaurant.app/book`). Antarmuka pelanggan dibangun sebagai target terpisah pada Xcode dengan arsitektur MVVM, terdiri dari empat layar utama yaitu detail pemesanan, detail tamu, barcode pembayaran QRIS untuk tamu lebih dari 10 orang dengan countdown timer serta konfirmasi reservasi. Implementasi dirancang dengan batasan teknis berupa ukuran maksimal 15 MB tanpa transisi ke aplikasi penuh. Sistem menerapkan prinsip no data persistence di mana data reservasi langsung dikirim ke server melalui RESTful API dengan keamanan autentikasi JWT.

4 Hasil dan Pembahasan

A. Implementasi Sistem

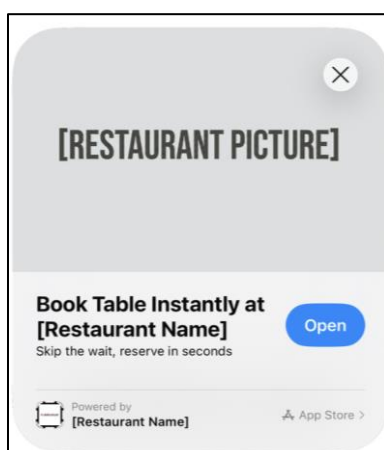
a) Implementasi Model-View-Model (MVVM)

<pre> struct BookingRequest: Encodable { let pax: Int let date: String let time: String let seating_area: String let is_smoking_area: Bool let is_allergy: Bool let occasion_id: Int? let ambience: [Int] let customer_request: CustomerRequest let price: Int let code: Int let payment_code: String struct CustomerRequest: Encodable { let name: String let email: String let phone: String } } </pre>	<pre> @Published var isLoading: Bool @Published var submitError: String? private let addBookingCase: AddBookingCaseProtocol init(addBookingCase: AddBookingCaseProtocol = AddBookingCase()) { self.addBookingCase = addBookingCase } func submit(bookingData: BookingData) async -> Bool { submitError = nil guard let date = bookingData.date, let time = bookingData.time else { submitError = "Missing date/time" return false } let request = BookingRequest(pax: bookingData.paxCount, date: dateStr, time: timeStr, seating_area: bookingData.seating_area, is_smoking_area: bookingData.is_smoking_area, is_allergy: bookingData.is_allergy, occasion_id: bookingData.occasion_id, ambience: bookingData.ambience, customer_request: .init(name: bookingData.guestName, email: bookingData.guestEmail, phone: bookingData.guestPhone), price: bookingData.price, code: bookingData.code, payment_code: bookingData.paymentCode ?? "") isLoading = true defer { isLoading = false } do { try await addBookingCase.execute(request) return true } catch { submitError = error.localizedDescription return false } } </pre>	<pre> struct BookingConfirmationView: View { @ObservedObject var bookingData: BookingData @StateObject private var vm = BookingViewModel() @State private var loadingPhase: LoadingPhase = .idle var onConfirm: (Bool) -> Void var onEdit: () -> Void private enum LoadingPhase { case idle, loading, success } var body: some View { VStack { ScrollView { // Display booking details } Button(action: { Task { loadingPhase = .loading let ok = await vm.submit(bookingData: bookingData) if ok { loadingPhase = .success onConfirm(true) } else { loadingPhase = .idle } } }) { Text("Confirm Booking") } .disabled(vm.isLoading) Button(action: { onEdit() }) { Text("Edit Booking") } if let err = vm.submitError { Text(err).foregroundColor(.red) } } } } </pre>
--	---	--

Gambar 11 Lapisan model (bookingrequest DTO), lapisan viewmodel (bookingviewmodel) dan lapisan view (bookingconfirmationview)

Gambar 11 (kiri) menunjukkan struktur BookingRequest DTO yang mengimplementasikan Encodable protocol untuk komunikasi dengan backend REST API. DTO menggunakan snake_case untuk compatibility dengan Golang backend dan nested CustomerRequest untuk encapsulation data pelanggan. Gambar 11 (tengah) menunjukkan implementasi BookingViewModel dengan @Published properties untuk reactive state management. ViewModel menggunakan dependency injection pattern dan async/await untuk asynchronous operations. Gambar 11 (kanan) menunjukkan lapisan View dengan data binding menggunakan @StateObject dan @ObservedObject. Async operation dibungkus dalam Task untuk concurrent execution. View secara otomatis ter-update ketika ViewModel mengubah state melalui @Published properties (isLoading, submitError).

b) Implementasi App Clip



Gambar 12 Tampilan kartu app clip

Gambar 12 menampilkan kartu App Clip yang muncul ketika pengguna memindai kode. Kartu ini menampilkan informasi restoran beserta tombol "Open" untuk mengakses halaman reservasi.

c) Implementasi Antarmuka Pelanggan

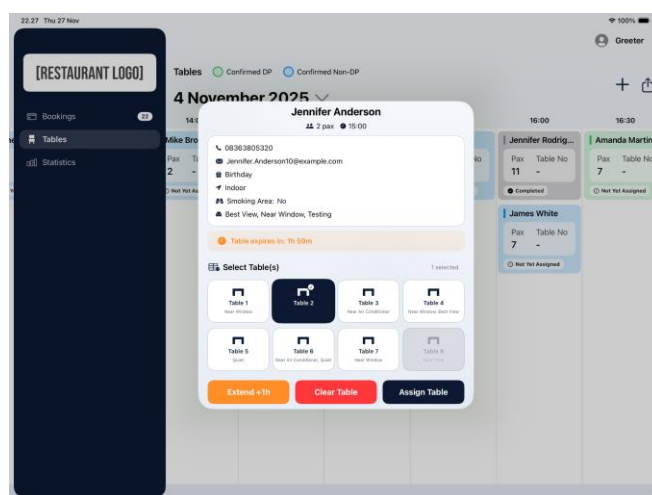
Antarmuka pelanggan dirancang untuk memfasilitasi proses reservasi dari awal hingga konfirmasi pembayaran. Alur antarmuka terdiri dari satu halaman utama.

Gambar 13 Antarmuka halaman *booking information* untuk *input data reservasi*

Gambar 13 menampilkan antarmuka halaman *Booking Information* sebagai tahap pertama dalam alur reservasi pelanggan. Pada halaman ini, pelanggan mengisi informasi dasar reservasi yang meliputi tanggal kunjungan, waktu kedatangan, dan jumlah tamu. Sistem mengimplementasikan validasi ketersediaan meja secara otomatis berdasarkan data yang dimasukkan, sehingga pelanggan hanya dapat memilih slot waktu yang tersedia. Mekanisme ini berfungsi untuk mencegah terjadinya *overbooking* dan memastikan kapasitas restoran tidak melebihi batas yang ditentukan.

d) Implementasi Antarmuka Admin

Antarmuka pelanggan dirancang untuk memfasilitasi proses reservasi dari awal hingga konfirmasi pembayaran. Alur antarmuka terdiri dari satu halaman utama.



Gambar 14 Antarmuka halaman *tables* dengan visualisasi *booking card* dan panel pengelolaan meja

Gambar 14 menampilkan antarmuka halaman *Tables* yang menyajikan visualisasi *Booking Card* untuk setiap reservasi aktif. Setiap kartu menampilkan informasi nama tamu, jumlah tamu, dan nomor meja yang ditetapkan, sehingga admin dapat memantau alokasi meja secara keseluruhan. Ketika admin mengklik salah satu *Booking Card*, sistem menampilkan *pop-up panel* yang menyediakan empat opsi pengelolaan yaitu *Assign Table* untuk menetapkan nomor meja, *Edit* untuk mengubah nomor meja, *Extend* untuk memperpanjang waktu reservasi, dan *Clear* untuk mengosongkan meja.

Selain itu, halaman ini juga menyediakan fitur *Walk-in* yang memungkinkan admin untuk menambahkan tamu yang datang tanpa reservasi sebelumnya. Fungsionalitas ini memberikan fleksibilitas dalam pengelolaan meja secara *real-time*.

B. Pengujian Sistem

a) Metode Pengujian (*Task-Based Usability Testing*)

Usability didefinisikan sebagai sejauh mana sistem dapat digunakan untuk mencapai tujuan dengan efektivitas, efisiensi, dan kepuasan [23], [24]. Evaluasi sistem menggunakan metode *task-based usability testing* yang memungkinkan observasi langsung terhadap perilaku pengguna dalam kondisi nyata [25]. Pengujian dilakukan *on-site* di restoran menggunakan iPhone 15 (iOS 18.0.1) dan iPad Pro 13-inch M4 (iOS 18.7.1) melalui TestFlight dengan durasi 15-30 menit per responden. Tugas dinyatakan berhasil apabila partisipan dapat menyelesaikan secara mandiri tanpa bantuan, dan gagal apabila tidak dapat diselesaikan meskipun diberikan kesempatan bantuan. Data dikumpulkan melalui observasi untuk mengukur *task completion rate* dan wawancara untuk memperoleh umpan balik kualitatif terkait perbaikan sistem.

b) Responden Pengujian

Partisipan dalam pengujian ini terdiri dari 7 responden yang terbagi menjadi dua kelompok berdasarkan peran pengguna dalam sistem. Tabel 1 menyajikan profil responden kelompok pelanggan dengan kode C sebanyak 3 responden, sedangkan Tabel 2 menyajikan profil responden kelompok admin dengan kode A sebanyak 4 responden.

Tabel 1 Profil responden pelanggan

Kode	Jenis Kelamin	Usia
C1	Laki-laki	29 tahun
C2	Laki-laki	36 tahun
C3	Perempuan	24 tahun

Tabel 2 Profil responden admin

Kode	Jenis Kelamin	Usia
A1	Laki-laki	31 tahun
A2	Laki-laki	34 tahun
A3	Laki-laki	37 tahun
A4	Perempuan	26 tahun

Kriteria inklusi untuk responden pelanggan meliputi: (1) berusia 20-40 tahun, (2) memiliki pengalaman menggunakan aplikasi reservasi restoran, dan (3) memiliki frekuensi makan di restoran minimal 2-3 kali per minggu. Kriteria inklusi untuk responden admin meliputi: (1) berusia 20-40 tahun, dan (2) memiliki pengalaman menggunakan sistem reservasi dari sisi administrator.

C. Hasil Pengujian

Usability testing dilaksanakan dengan skenario tugas yang disesuaikan berdasarkan peran pengguna. Responden antarmuka pelanggan menyelesaikan 7 tugas yang mencakup proses reservasi meja, sedangkan responden antarmuka admin menyelesaikan 17 tugas yang mencakup pengelolaan reservasi dan pembayaran. Setelah menyelesaikan seluruh tugas, responden diwawancarai untuk memperoleh umpan balik kualitatif dengan dua pertanyaan utama: (1) "Apakah ada kebingungan atau hal yang tidak jelas selama pengujian?", dan (2) "Secara keseluruhan, apakah ada saran untuk perbaikan aplikasi?"

a) Hasil Pengujian App Clip

Pengujian App Clip terintegrasi dalam pengujian antarmuka pelanggan, khususnya pada tugas TC-01 pada Tabel 3. Seluruh responden (3 dari 3) berhasil mengakses aplikasi melalui App Clip dengan memindai kode menggunakan kamera perangkat tanpa mengalami kendala. Hal ini menunjukkan bahwa implementasi App Clip telah memenuhi tujuan desain yaitu menyediakan akses cepat ke fitur reservasi tanpa perlu mengunduh aplikasi lengkap.

b) Hasil Pengujian Antarmuka Pelanggan

Hasil pengujian menunjukkan bahwa seluruh tugas berhasil diselesaikan oleh semua responden tanpa memerlukan bantuan. Tabel 3 menyajikan hasil *task completion rate* (TCR) untuk setiap tugas.

Tabel 3 Hasil pengujian antarmuka pelanggan

Kode	Skenario	Berhasil	Gagal	TCR
TC-01	Akses App Clip via kamera	3	0	100%
TC-02	Isi reservasi (skenario <i>full booked</i>)	3	0	100%
TC-03	Isi reservasi (skenario tersisa 10 meja)	3	0	100%
TC-04	Isi reservasi (skenario berhasil)	3	0	100%
TC-05	Konfirmasi dan isi detail tamu	3	0	100%
TC-06	Pembayaran <i>down payment</i> QRIS	3	0	100%
TC-07	Lihat status reservasi	3	0	100%

Meskipun seluruh tugas berhasil diselesaikan, responden memberikan saran perbaikan berdasarkan hasil wawancara. Tabel 4 merangkum saran perbaikan dari responden pelanggan.

Tabel 4 Saran perbaikan dari responden pelanggan

No	Aspek	Saran Perbaikan
1.	Validasi pembayaran	Fitur unggah bukti transfer agar pelanggan yakin pembayaran telah diterima

c) Hasil Pengujian Antarmuka Admin

Hasil pengujian menunjukkan bahwa seluruh tugas berhasil diselesaikan oleh semua responden. Tabel 5 menyajikan hasil *task completion rate* (TCR) untuk setiap tugas.

Tabel 5 Hasil pengujian antarmuka admin

Kode	Skenario	Berhasil	Gagal	TCR
TA-01	<i>Login</i>	4	0	100%
TA-02	Konfirmasi pembayaran gagal	4	0	100%
TA-03	Informasikan pelanggan via WhatsApp	4	0	100%
TA-04	Konfirmasi pembayaran berhasil	4	0	100%
TA-05	Tetapkan meja untuk reservasi	4	0	100%
TA-06	Tetapkan meja untuk <i>walk-in</i>	4	0	100%
TA-07	Perpanjang waktu meja 1 jam	4	0	100%
TA-08	Ubah meja reservasi	4	0	100%
TA-09	Kosongkan semua meja <i>walk-in</i>	4	0	100%
TA-10	Kosongkan meja reservasi	4	0	100%
TA-11	Unduh data reservasi (PDF)	4	0	100%
TA-12	Lihat ringkasan <i>highlights</i>	4	0	100%
TA-13	Perluas <i>highlights</i> untuk melihat grafik	4	0	100%
TA-14	Unduh grafik <i>highlights</i> (PDF)	4	0	100%
TA-15	Lihat <i>charts</i>	4	0	100%
TA-16	Unduh <i>chart</i> (PDF)	4	0	100%
TA-17	<i>Logout</i>	4	0	100%

Meskipun seluruh tugas berhasil diselesaikan, responden memberikan beberapa saran perbaikan berdasarkan hasil wawancara. Tabel 6 merangkum saran perbaikan dari responden admin.

Tabel 6 Saran perbaikan dari responden admin

No	Aspek	Saran Perbaikan
1.	Unggah bukti pembayaran	Fitur validasi bukti transfer untuk memudahkan kasir tanpa akses bank langsung
2.	<i>Edit</i> detail reservasi	Fitur <i>edit</i> reservasi setelah pengajuan untuk mengakomodasi perubahan rencana
3.	<i>Input</i> detail <i>walk-in</i>	<i>Input</i> nama dan jumlah tamu <i>walk-in</i> pada <i>Booking Card</i>

D. Pembahasan

a) Analisis Hasil Pengujian

Hasil *usability testing* menunjukkan *task completion rate* (TCR) 100% pada kedua antarmuka, melampaui *benchmark* 78%, yaitu median dari 1.189 tugas *usability* di 115 studi [26] yang mengindikasikan kinerja di atas rata-rata. Capaian ini memvalidasi *Technology Acceptance Model* bahwa *perceived ease of use* atau persepsi sistem dapat digunakan tanpa upaya besar [27], merupakan penentu utama adopsi teknologi. Penghapusan *friction point* instalasi melalui App Clip meningkatkan *perceived ease of use*, sejalan dengan *effort expectancy theory* [9] bahwa pengurangan kompleksitas proses meningkatkan niat perilaku. Keberhasilan 100% pada akses App Clip (TC-01) mengonfirmasi bahwa teknologi *instant app* mengatasi hambatan instalasi penyebab 77,9% pembatalan transaksi [10].

Penyelesaian 7 tugas pelanggan dan 17 tugas admin mengindikasikan sistem memenuhi Nielsen's *usability heuristics* [28] yaitu *consistency and standards* dimana MVVM memastikan prediktabilitas antarmuka, *user control and freedom* dimana QRIS kondisional memberikan fleksibilitas, serta *aesthetic minimalist design* dimana App Clip membatasi fitur esensial saja. TCR 100% menunjukkan tidak ada masalah *usability*, sesuai ISO 9241-11 yang menetapkan TCR sebagai ukuran fundamental efektivitas [23]. Keberhasilan *multi-role* admin (manajer, staf *finance*, *greeter*) menunjukkan sistem mudah dipelajari oleh berbagai tipe pengguna, mendukung efektivitas MVVM dalam menjaga konsistensi antarmuka melalui *separation of concerns* [16].

Meskipun TCR 100%, umpan balik kualitatif mengungkap tiga area perbaikan: (1) pengunggahan bukti pembayaran untuk validasi transaksi, (2) pengeditan reservasi setelah pengajuan untuk mengakomodasi perubahan rencana, dan (3) pencatatan detail tamu *walk-in* untuk kelengkapan data analitik. Szmydt [29] menekankan bahwa keberhasilan aplikasi *mobile* ditentukan oleh *perceived quality of use*, sehingga umpan balik ini penting untuk penyempurnaan iteratif. Temuan memvalidasi App Clip sebagai solusi layak mengatasi hambatan instalasi tanpa mengorbankan *usability*, yang berkontribusi pada *body of knowledge* teknologi *instant app* dalam sektor *hospitality*.

b) Kesesuaian Dengan Kerangka CBL

Tabel 7 Kesesuaian tahapan penelitian dengan kerangka CBL

Fase CBL	Kerangka	Implementasi dalam Penelitian
Engage	Big Idea	Efisiensi Pelayanan Restoran
	Essential Question	Bagaimana teknologi <i>mobile</i> dapat meningkatkan aksesibilitas dan efisiensi sistem reservasi?
	Challenge	Sistem reservasi tanpa instalasi aplikasi penuh
Investigate	Guiding Questions	(1) Reservasi cepat tanpa instalasi? (2) Pencegahan <i>overbooking</i> otomatis? (3) Pengelolaan reservasi <i>real-time</i> ?
	Guiding Activities	Survei <i>stakeholder</i> , analisis App Clip, perancangan MVVM, implementasi sistem
	Guiding Resources	Dokumentasi Apple Developer, referensi akademik, <i>stakeholder</i>
Act	Solution	App Clip untuk pelanggan, aplikasi admin untuk pengelolaan reservasi
	Implementation	Perancangan UML dan implementasi SwiftUI, Golang, PostgreSQL
	Evaluation	<i>Usability testing</i> 7 responden (4 admin, 3 pelanggan)

Hasil *task completion rate* 100% menunjukkan solusi berhasil menjawab *challenge* yang dirumuskan, sementara umpan balik kualitatif yaitu pengunggahan bukti pembayaran, pengeditan reservasi setelah pengajuan, dan pencatatan detail tamu *walk-in* memberikan landasan iterasi berikutnya sesuai prinsip CBL yang menekankan pembelajaran berkelanjutan melalui refleksi dan perbaikan [14].

5 Kesimpulan

Penelitian ini berhasil menjawab tiga *research questions* yang dirumuskan. Pertama (RQ1), penelitian ini merancang arsitektur sistem reservasi restoran berbasis iOS menggunakan teknologi App Clip menggunakan pola MVVM yang terintegrasi dengan *dashboard* admin secara *real-time*, dimana pemisahan *Model-View-ViewModel* memastikan konsistensi antarmuka dan memfasilitasi *dual-interface* (pelanggan dan admin) dalam satu ekosistem aplikasi. Kedua (RQ2), penelitian ini berhasil mengimplementasikan fitur-fitur utama sistem reservasi: App Clip sebagai akses instan untuk eliminasi *friction* instalasi, pembayaran uang muka kondisional melalui QRIS untuk tamu lebih dari 10 orang guna meminimalkan *no-show*, pemilihan meja berdasarkan preferensi *ambience*, dan mekanisme *auto-prevention overbooking*. Ketiga (RQ3), *usability testing* berbasis *task completion* terhadap tujuh responden (4 admin, 3 pelanggan) menunjukkan *task completion rate* 100%, melampaui rata-rata *benchmark* 78% [26], mengonfirmasi tingkat *usability* tinggi pada kedua antarmuka. Meskipun demikian, penelitian ini memiliki beberapa keterbatasan. Pertama, jumlah responden terbatas (7 orang). Kedua, evaluasi *usability* hanya mengukur *task completion rate* dan umpan balik kualitatif tanpa metrik kuantitatif lain seperti *task time* atau *error rate*. Ketiga, fitur sistem masih berfokus pada alur reservasi utama dan belum mencakup kebutuhan operasional tambahan yang teridentifikasi dari umpan balik responden yaitu (1) pengunggahan bukti pembayaran, (2) pengeditan reservasi setelah pengajuan, dan (3) pencatatan detail tamu *walk-in*. Pengembangan selanjutnya disarankan untuk: (1) melibatkan responden lebih banyak dan beragam untuk validasi lebih luas, (2) memperluas metrik evaluasi *usability* dengan *task time* dan *error rate* sesuai ISO 9241-11 [23], (3) mengimplementasikan tiga fitur operasional tambahan yang teridentifikasi, dan (4) melakukan evaluasi performa sistem pada kondisi *concurrent users* yang tinggi untuk memastikan skalabilitas.

Referensi

- [1] Badan Pusat Statistik, “Statistik Penyediaan Makanan dan Minuman 2023.” [Online]. Available: <https://www.bps.go.id/en/publication/2024/12/23/f2c7743c4712aaea4abf694/statistik-penyediaan-makanan-dan-minuman-2023.html>
- [2] G. M. Thompson and R. J. Kwortnik, “*Pooling Restaurant Reservations to Increase Service Efficiency*,” *J. Serv. Res.*, Vol. 10, No. 4, pp. 335–346, 2008, DOI: 10.1177/1094670508314267.
- [3] D. Roy, E. Spiliotopoulou, and J. de Vries, “*Restaurant Analytics: Emerging Practice and Research Opportunities*,” *Prod. Oper. Manag.*, Vol. 31, No. 10, pp. 3687–3709, 2022, DOI: 10.1111/poms.13809.
- [4] L. Bujalance-López, L. González-Serrano, M. P. Lechuga Sancho, and P. Talon-Ballester, “*Restaurant Revenue Management: A Systematic Literature Review and Future Challenges*,” *Br. Food J.*, Vol. 127, No. 6, pp. 2169–2196, 2025, DOI: 10.1108/BFJ-08-2024-0816.
- [5] M. B. Ramadhan, N. R. Triana, and R. D. Sheva, “Perancangan Sistem informasi Reservasi Restoran berbasis *Web* menggunakan Metode *Waterfall* (Studi Kasus : Rumah Makan Joglo (RMJ)),” Vol. 4, No. 6, pp. 216–229, 2025.
- [6] Asosiasi Penyelenggara Jasa Internet Indonesia, “Survei Penetrasi Internet Indonesia 2024.” [Online]. Available: <https://apjii.or.id>
- [7] Asosiasi Penyelenggara Jasa Internet Indonesia, “Pengguna Internet RI 2025 Tembus 229,4 Juta, Gen Z mendominasi.” [Online]. Available: <https://www.cloudcomputing.id/berita/pengguna-internet-ri-2025-229-4-juta>
- [8] Bank Indonesia, “Statistik Sistem Pembayaran dan Pengelolaan Uang Rupiah: Data Transaksi QRIS Juli 2024.” [Online]. Available: <https://www.bi.go.id>
- [9] R. Palau-Saumell, S. Forgas-Coll, J. Sánchez-García, and E. Robres, “*User Acceptance of Mobile Apps for Restaurants: An Expanded and Extended UTAUT-2*,” *Sustain.*, Vol. 11, No. 4, pp. 1–24, 2019, DOI: 10.3390/su11041210.
- [10] Heady, “*Mobile Customer Experience Issues Highlight use Cases for App Clips*,” Feb. 2021. [Online]. Available: <https://www.heady.io/blog/market-study-mobile-customer-experience-issues-highlight-use-cases-for-ios-app-clips>
- [11] M. Padigar, Y. Li, and C. N. Manjunath, “*‘Good’ and ‘Bad’ Frictions in Customer Experience: Conceptual Foundations and Implications*,” *Psychol. Mark.*, Vol. 42, No. 1, pp. 21–43, 2025,

<http://sistemasi.ftik.unisi.ac.id>

- DOI: 10.1002/mar.22111.
- [12] Apple Inc., “*App Clips*.” [Online]. Available: <https://developer.apple.com/documentation/appclip>
- [13] S. E. Gallagher and T. Savage, “*Challenge-based Learning in Higher Education: An Exploratory Literature Review*,” *Teach. High. Educ.*, Vol. 28, No. 6, pp. 1135–1157, 2023, DOI: 10.1080/13562517.2020.1863354.
- [14] K. Doulougeri, J. D. Vermunt, G. Bombaerts, and M. Bots, “*Challenge-based Learning Implementation in Engineering Education: A Systematic Literature Review*,” *J. Eng. Educ.*, Vol. 113, No. 4, pp. 1076–1106, 2024, DOI: 10.1002/jee.20588.
- [15] M. Nichols, K. Cator, and M. Torre, *Challenge based Learner User Guide*. Digital Promise, 2016. [Online]. Available: https://www.challengebasedlearning.org/wp-content/uploads/2019/02/CBL_Guide2016.pdf
- [16] D. Indrawan, D. S. Kusumo, and S. Y. Puspitasari, “*Analysis of the Implementation of MVVM Architecture Pattern on Performance of iOS Mobile-based Applications*,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, Vol. 8, No. 1, pp. 59–65, 2023, DOI: 10.29100/jupi.v8i1.3293.
- [17] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner’s Approach, 9th editio*. New York, NY: McGraw-Hill Education, 2020.
- [18] Apple Inc., “*Creating an App Clip with Xcode*.” [Online]. Available: <https://developer.apple.com/documentation/appclip/creating-an-app-clip-with-xcode>
- [19] A. A. A. Donovan and B. W. Kernighan, *The Go Programming Language*. Boston, MA: Addison-Wesley, 2015.
- [20] J. R. Z. Wei and X. Gao, “*Automated Code Transformation for Context Propagation in Go*,” in *29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, Athens, Greece: ACM (Association for Computing Machinery), 2021. DOI: 10.1145/3468264.3473122.
- [21] PostgreSQL Global Development Group, “*PostgreSQL 16 Documentation*.” [Online]. Available: <https://www.postgresql.org/docs/16/>
- [22] C. Pacheco, I. Garcia, and M. Reyes, “*Requirements Elicitation Techniques: A Systematic Literature Review based on the Maturity of the Techniques*,” *IET Softw.*, Vol. 12, No. 4, pp. 365–378, 2018, DOI: 10.1049/iet-sen.2017.0144.
- [23] International Organization for Standardization (ISO), “*Ergonomics of Human-System Interaction — Part 11: Usability: Definitions and Concepts*,” 2018. [Online]. Available: <https://www.iso.org/standard/63500.html>
- [24] J. Rubin and D. Chisnell, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, 2nd editio*. Indianapolis, IN: Wiley, 2008.
- [25] P. Weichbroth, “*Usability of Mobile Applications: A Systematic Literature Study*,” *IEEE Access*, Vol. 8, pp. 55563–55577, 2020, DOI: 10.1109/ACCESS.2020.2981892.
- [26] J. R. Lewis and J. Sauro, “*Usability and User Experience: Design and Evaluation*,” in *Handbook of Human Factors and Ergonomics, 5th editio.*, G. Salvendy and W. Karwowski, Eds., Hoboken, NJ: Wiley, 2021, pp. 972–1015.
- [27] F. D. Davis, “*Perceived Usefulness, Perceived Ease of use, and User Acceptance of Information Technology*,” *MIS Quarterly*, Vol. 13, No. 3, pp. 319–340, 1989. [Online]. Available: <https://www.jstor.org/stable/249008>
- [28] J. Nielsen, “*Enhancing the Explanatory Power of Usability Heuristics*,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*, Boston, MA, USA, 1994, pp. 152–158. [Online]. Available: <https://doi.org/10.1145/191666.191729>
- [29] M. Szmydt, “*Usability Testing of Mobile Applications: A Methodological Framework*,” *Applied Sciences*, Vol. 14, No. 5, p. 1792, 2024. [Online]. Available: <https://doi.org/10.3390/app14051792>