

BAB II

LANDASAN TEORI

2.1 Rancang Bangun

Rancang dapat diartikan sebagai aktivitas yang bertujuan untuk mendesain suatu sistem baru guna menjawab permasalahan yang dihadapi pengguna melalui pemilihan alternatif sistem yang paling tepat. Sementara itu, istilah “bangun” berasal dari kata “pembangunan” yang mencerminkan proses menciptakan sistem baru ataupun melakukan penggantian dan perbaikan terhadap sistem yang sudah ada, baik secara menyeluruh maupun sebagian. Oleh karena itu, rancang bangun dapat dipahami sebagai kegiatan menerjemahkan hasil analisis ke dalam bentuk perangkat lunak, kemudian merealisasikan sistem baru atau menyempurnakan sistem yang telah tersedia (Sitanggal et al, 2022).

2.2 Sistem Informasi

Teori sistem merupakan sekumpulan pernyataan yang digunakan untuk membantu orang dalam memahami tentang sistem sehingga para praktisi memperoleh penjelasan yang kuat dan dapat menginterpretasikannya dengan tepat (Heryana, 2021).

Informasi adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan, sebagaimana perannya, pengguna membuat keputusan yang lebih baik sebagai kuantitas dan kualitas dari peningkatan informasi (Sangga Rasefta & Esabella, 2020).

Sistem informasi adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, dan mengolah serta menyimpan data, dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan (Taufik dkk., 2022).

2.3 Aplikasi *Mobile Android*

Program aplikasi merupakan perangkat lunak yang berfungsi untuk mendukung penyelesaian tugas-tugas tertentu. Salah satu bentuknya adalah aplikasi *mobile*, yaitu perangkat lunak yang dirancang khusus agar dapat dijalankan pada perangkat tablet maupun *smartphone*. Aplikasi *mobile* memiliki karakteristik

tersendiri yang membedakannya dari jenis aplikasi lainnya. Aplikasi *mobile* yang dirancang dengan baik diharapkan mampu menyajikan informasi serta layanan kepada pengguna secara lebih efisien dan praktis (Karim & Adriansyah, 2022).

2.4 Dart

Dart merupakan bahasa pemrograman terstruktur dan bersifat *open source* yang dikembangkan untuk membangun aplikasi *web* berbasis *browser* dengan tingkat kompleksitas yang tinggi. Aplikasi yang ditulis dengan *Dart* dapat dijalankan langsung pada *browser* yang mendukung kode *Dart*, maupun melalui proses kompilasi ke dalam bahasa *JavaScript*. Bahasa *Dart* memiliki sintaks yang mirip dengan *Java*, *C#*, serta *JavaScript*, bersifat berbasis kelas, mendukung *optional typing*, serta berjalan dalam mode *single-threaded*. *Dart* juga memiliki model konkurensi bernama *isolate* yang memungkinkan eksekusi paralel. Selain dapat dijalankan pada *browser* dan dikompilasi ke *JavaScript*, *Dart* juga dapat dijalankan melalui baris perintah menggunakan *Dart Virtual Machine (Dart VM)*, sehingga memungkinkan pengembangan sisi klien maupun sisi server dalam satu bahasa pemrograman yang sama. Salah satu tujuan utama dari *Dart* adalah menghadirkan sintaks yang mudah dipahami oleh pengembang, dengan struktur sederhana seperti fungsi *main()* sebagai titik awal program (Sofi & Dharmawan, 2022).

2.5 Flutter

(Setiawan dkk., 2022). *Flutter* merupakan sebuah *framework* bersifat *open source* yang dikembangkan oleh *Google* untuk membangun antarmuka pengguna (*user interface*) pada aplikasi *Android* maupun *iOS*. Saat ini, *Flutter* telah banyak diadopsi oleh perusahaan besar sebagai teknologi pengembangan aplikasi lintas *platform*. *Flutter* memiliki sejumlah keunggulan, antara lain:

1. *Cross-Platform*, yaitu kemampuan untuk menjalankan aplikasi pada berbagai *platform* berbeda. Dengan *Flutter*, pengembang dapat membangun aplikasi *Android* dan *iOS* sekaligus, bahkan juga aplikasi berbasis *web* dan *desktop*, sehingga menghemat waktu tanpa perlu mempelajari bahasa pemrograman *native* tiap *platform*.

2. *Fast Development (Hot Reload)*, yaitu fitur yang memungkinkan perubahan kode program diterapkan secara cepat hanya dengan menekan *hot reload*, sehingga aplikasi diperbarui dalam hitungan detik. Fitur ini sangat membantu dalam mempercepat proses pengembangan.
3. *Beautiful UI*, karena *Flutter* menyediakan kemudahan dalam membangun tampilan antarmuka melalui konsep *widget*. Seluruh elemen antarmuka disusun menggunakan *widget* yang dapat dikustomisasi sesuai kebutuhan, dengan dukungan dua set utama yaitu *Material Design* (untuk Android) dan *Cupertino* (untuk iOS).

2.6 Visual Studio Code

Visual Studio Code merupakan perangkat lunak editor kode sumber yang bersifat ringan namun memiliki performa tinggi, serta dapat dijalankan pada perangkat *desktop*. Aplikasi ini banyak dimanfaatkan dalam proses penulisan kode program karena mendukung berbagai bahasa pemrograman, seperti *JavaScript*, *HTML*, *CSS*, *PHP*, *Python*, *C++*, dan lain sebagainya. *Visual Studio Code* juga bersifat lintas platform sehingga dapat dijalankan pada sistem operasi *Windows*, *macOS*, maupun *Linux*. Selain itu, tersedia pula fitur *Live Share* yang memungkinkan beberapa pengembang berkolaborasi pada satu proyek secara bersamaan meskipun berada di lokasi yang berbeda (Nanda Syarif dkk., 2023).

2.7 Firebase

Firebase merupakan layanan *cloud service provider* sekaligus *backend as a service* yang dikembangkan oleh *Google*. Platform ini hadir sebagai solusi untuk mempermudah proses pengembangan aplikasi berbasis *mobile* maupun *web*, dengan salah satu fitur utamanya berupa *Realtime Database* yang memungkinkan sinkronisasi data secara langsung antara klien dan server. (Andrianto & Haris Munandar, 2022)

2.8 Prototype

Model *prototype* merupakan salah satu metode pengembangan perangkat lunak yang menekankan pada pembuatan *mockup* atau model aplikasi. Pendekatan ini dinilai efektif ketika pengguna mengalami kesulitan dalam menjelaskan kebutuhan sistem secara rinci sesuai dengan keinginan mereka, sehingga model

yang dibuat dapat membantu memberikan gambaran awal terhadap aplikasi yang akan dikembangkan (Meisak dkk., 2022).

2.9 Business Process Managemen

Konsep *Business Process Management* (BPM) merupakan pendekatan manajemen kinerja yang efektif bagi organisasi yang berorientasi pada proses. BPM dapat dipahami sebagai disiplin manajerial sekaligus seperangkat teknologi yang digunakan untuk mengelola proses bisnis yang melibatkan berbagai unit, keahlian, lokasi, serta tingkatan dalam organisasi. Penerapan BPM menuntut pemahaman mendalam terhadap tahapan proses bisnis, pihak-pihak yang terlibat, aliran informasi, serta teknologi yang digunakan agar pelaksanaannya berjalan optimal. Dalam mengimplementasikan BPM dan meningkatkan kualitas proses, data, serta informasi, diperlukan adanya inventarisasi pengetahuan yang berfungsi untuk mendefinisikan proses organisasi. Selain itu, dibutuhkan pula perangkat pendukung yang mampu menjamin fleksibilitas dan kemampuan adaptasi BPM secara terpadu, yang mencakup strategi bisnis, proses bisnis, strategi teknologi informasi (TI), infrastruktur TI, dan mekanisme proses yang saling terkait (Rasendriya Aniko dkk., 2024)

2.10 Database

Database atau basis data merupakan gabungan dari dua istilah, yaitu *basis* dan *data*. Istilah *basis* dapat diartikan sebagai tempat penyimpanan atau pusat pengumpulan, sedangkan *data* merujuk pada sekumpulan fakta yang menggambarkan objek atau kejadian nyata di dunia, seperti manusia, barang, hewan, peristiwa, maupun konsep tertentu. Fakta tersebut dapat direpresentasikan dalam berbagai bentuk, antara lain huruf, angka, simbol, teks, gambar, suara, atau kombinasi dari semuanya. Dengan demikian, basis data dapat dipahami sebagai sekumpulan informasi yang tersusun secara sistematis dan terorganisir di dalam komputer untuk mendukung proses pengolahan data dan pengambilan keputusan. Dalam konteks sistem informasi, basis data berperan penting sebagai komponen utama yang menyimpan, mengelola, serta menyediakan akses cepat dan efisien terhadap data yang dibutuhkan oleh pengguna atau aplikasi (Gede dkk., 2022)

2.11 Figma

Figma adalah salah satu perangkat lunak yang digunakan oleh desainer UI/UX untuk merancang tampilan visual pada aplikasi *mobile* maupun situs *web*. Berbeda dengan *Adobe Photoshop*, *Figma* memiliki keunggulan utama dalam fitur kolaborasi, yang memungkinkan beberapa anggota tim bekerja secara bersamaan pada satu proyek desain. Melalui fitur tersebut, pengguna dapat memberikan komentar, masukan, maupun melakukan penyuntingan secara *real time*. Selain itu, *Figma* juga memiliki kemampuan untuk menyimpan setiap perubahan desain secara otomatis selama perangkat terhubung ke internet, sehingga meminimalkan risiko kehilangan data dan meningkatkan efisiensi kerja tim (Senubekti dkk, 2024).

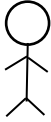
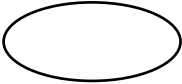
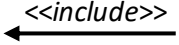
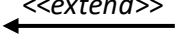
2.12 Unified Modelling Language

Unified Modeling Language (UML) merupakan salah satu alat bantu pemodelan yang banyak digunakan dalam pengembangan sistem berbasis *object-oriented*. *UML* lahir dari penggabungan beberapa metode pemodelan, yaitu *Booch*, *Object Modeling Technique (OMT)*, serta *Object-Oriented Software Engineering (OOSE)*. Dengan menggunakan metode ini, proses analisis dan perancangan dilakukan secara iteratif, mencakup identifikasi kelas dan objek, penentuan hubungan semantik antar objek dan kelas, hingga perincian antarmuka serta tahap implementasi (Voutama, 2022). Adapun diagram tersebut diantaranya adalah *Class*, *use case*, dan *activity diagram*.

1. Use Case Diagram

Use case adalah representasi fungsi suatu sistem yang digambarkan dari sudut pandang pengguna. Melalui *use case*, dapat dijelaskan proses yang dilakukan sistem beserta komponen-komponennya. Model ini menggunakan skenario yang berisi urutan langkah-langkah interaksi antara pengguna dan sistem. Dengan demikian, *use case* berperan dalam mengidentifikasi fungsionalitas sistem, interaksi pengguna dengan sistem, serta hubungan antara pengguna dan fitur yang tersedia pada sistem (Setiyani, 2021). Tabel di bawah ini menjelaskan komponen-komponen yang ada pada *use case diagram*.

Tabel 2.1 *Use Case Diagram*

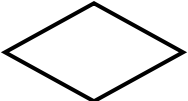
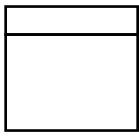
Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use Case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2. Activity Diagram

Activity Diagram merupakan salah satu komponen penting dalam *UML* yang digunakan untuk memodelkan aspek dinamis dari suatu sistem. Diagram ini mampu menggambarkan logika prosedural, proses bisnis, maupun alur kerja dengan lebih jelas. Tujuan utama dari *activity diagram*

adalah merepresentasikan perilaku dinamis sistem melalui visualisasi aliran aktivitas atau pesan dari satu kegiatan ke kegiatan lainnya(Annisa, 2020). Tabel di bawah ini menjelaskan komponen-komponen yang ada pada *activity diagram*.

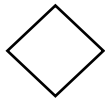
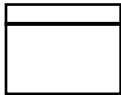
Tabel 2.2 Activity Diagram

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Percabangan / <i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
	Penggabungan / <i>Join</i>	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

3. Class Diagram

Class Diagram merupakan inti dari proses pemodelan objek. *Tool* yang membantu dalam visualisasi struktur kelas dari suatu *system*. Hal ini disebabkan karena *class* adalah deskripsi kelompok obyek-obyek dengan properti, perilaku (operasi) dan relasi yang sama. Disamping itu *class diagram* bisa memberikan panduan (Arif Novianto & Hari Purwanto, 2022). Tabel di bawah ini menjelaskan komponen-komponen yang ada pada *Class Diagram*.

Tabel 2.3 *Class Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Generalization</i>	Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		<i>Class</i>	Himpunan objek yang terbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang tidak mandiri.

7.	_____	<i>Associaton</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
----	-------	-------------------	--

2.13 Penelitian Terdahulu

Tabel 2.4 Penelitian Terdahulu

No	Judul	Peneliti, dan Tahun	Tujuan Penelitian	Hasil dan Metode Yang Digunakan	Saran atau Kelemahan	Kesenjangan Penelitian
1	Rancang Bangun Aplikasi Mobile Penjadwal Perkuliahan Dengan Firebase Dengan Realtime Notification	Yulia Darnita, Muntahanah (2021)	Mengembangkan aplikasi penjadwalan perkuliahan dengan notifikasi real-time kepada mahasiswa	Aplikasi berhasil memberikan notifikasi jadwal perkuliahan secara real-time kepada mahasiswa. Metode: <i>Rapid Application Development (RAD)</i>	Pelu pengujian lebih lanjut untuk memastikan kompatibilitas dengan berbagai perangkat	Penelitian ini berfokus pada bidang pendidikan, sedangkan penelitian ini pada kuliner (pemesanan makanan). Selain itu, penelitian ini menggunakan metode RAD, sementara penelitian saya menggunakan Prototype untuk memungkinkan interaksi langsung dengan pengguna selama pengembangan.
2	Rancang Bangun Aplikasi Mobile Pengenalan Huruf Jawa (Aksara	Astuti, Romawati, &	Mengembangkan aplikasi pembelajaran aksara Jawa	Aplikasi berhasil membantu pengguna dalam mengenali dan mempelajari	Diperlukan pengembangan lebih lanjut untuk	Penelitian ini berfokus pada media pembelajaran dengan metode Waterfall, sedangkan

No	Judul	Peneliti, dan Tahun	Tujuan Penelitian	Hasil dan Metode Yang Digunakan	Saran atau Kelemahan	Kesenjangan Penelitian
	Jawa) Berbasis Android	Widaningrum (2020)	berbasis Android untuk memudahkan pengguna dalam mempelajari huruf Jawa	aksara Jawa secara interaktif. Metode: <i>Waterfall</i>	menambahkan fitur latihan dan evaluasi	penelitian ini berfokus pada aplikasi transaksi kuliner dengan metode Prototype yang lebih fleksibel dan iteratif.
3	Rancang Bangun Aplikasi Mobile 'Pharmacy Src' Terhadap Pemetaan Lokasi Apotek di Kota Kediti	Wijaya, Cahyono, & Utomo (2023)	Mengembangkan aplikasi pemetaan lokasi apotek di kota Kediri untuk memudahkan masyarakat mencari apotek terdekat	Aplikasi berhasil memetakan lokasi apotek dan memberikan informasi detail kepada pengguna. Metode: <i>Waterfall</i>	Perlu integrasi dengan fitur navigasi untuk panduan rute ke apotek	Penelitian ini berada di bidang kesehatan, sedangkan penelitian ini di bidang kuliner. Penelitian ini memakai Waterfall, sementara penelitian saya menggunakan Prototype yang lebih cocok untuk menyesuaikan kebutuhan pengguna secara cepat.
4	Rancang Bangun Sistem Informasi Company Profile Berbasis Android	Muhamad Alda (2023)	Mengembangkan sistem informasi profil perusahaan berbasis android	Sistem informasi berhasil menampilkan profil perusahaan secara	Diperlukan evaluasi lebih lanjut terkait pengalaman	Penelitian ini berfokus pada profil perusahaan, sedangkan penelitian saya pada pemesanan

No	Judul	Peneliti, dan Tahun	Tujuan Penelitian	Hasil dan Metode Yang Digunakan	Saran atau Kelemahan	Kesenjangan Penelitian
	Menggunakan Metode Exrteme Programming (XP)		dengan metode Extreme Programming	efektif melalui platform Android. Metode: <i>Extreme Programming (XP)</i>	pengguna dan kinerja aplikasi	makanan. Selain itu, penelitian ini memakai metode XP, sedangkan penelitian ini menggunakan Prototype yang lebih sederhana dan berorientasi pada umpan balik pengguna.
5	Rancang Bangun Aplikasi dengan Perbandingan Metode K-Nearest Neighbor (KNN) dan Naive Bayes dalam Klasifikasi Penderita Penyakit Diabetes	Prasetya, & Sujatniko (2022)	Mengembangkan aplikasi untuk mengklasifikasikan penderita diabetes menggunakan metode KNN dan Naive Bayes	Aplikasi menunjukkan bahwa metode KNN memiliki akurasi lebih tinggi dibandingkan Naive Bayes dalam klasifikasi diabetes. Metode: <i>Data Mining (KNN & Naive Bayes)</i>	Perlu pengujian lebih lanjut dengan dataset yang lebih besar dan beragam untuk meningkatkan akurasi model	Penelitian ini menggunakan pendekatan data mining di bidang kesehatan, sementara penelitian ini berfokus pada pengembangan sistem pemesanan makanan dengan Prototype yang menitikberatkan pada interaksi pengguna dan desain antarmuka.

No	Judul	Peneliti, dan Tahun	Tujuan Penelitian	Hasil dan Metode Yang Digunakan	Saran atau Kelemahan	Kesenjangan Penelitian
6	Consumer behavior of mobile food ordering app users during COVID-19: dining attitudes, e-satisfaction, perceived risk, and continuance intention.	Timur, B., Oğuz, Y. E., & Yilmaz, V. (2023)	Menyelidiki perilaku konsumen pengguna aplikasi pemesanan makanan selama pandemi COVID-19, termasuk sikap terhadap makan, kepuasan elektronik. Persepsi risiko, dan niat untuk terus menggunakan aplikasi	Pandemi COVID-19 meningkatkan penggunaan aplikasi pemesanan makanan; kepuasan pengguna dipengaruhi oleh kualitas layanan dan persepsi risiko. Metode: <i>Structural Equation Modeling (SEM)</i>	Pengembang aplikasi disarankan untuk meningkatkan fitur keamanan dan higienitas dalam layanan mereka untuk mengurangi persepsi risiko konsumen	Penelitian ini fokus pada analisis perilaku konsumen, bukan pengembangan sistem. Penelitian ini berbeda karena fokus pada rancang bangun aplikasi pemesanan makanan menggunakan Prototype.

No	Judul	Peneliti, dan Tahun	Tujuan Penelitian	Hasil dan Metode Yang Digunakan	Saran atau Kelemahan	Kesenjangan Penelitian
7	Rancang Bangun Aplikasi Mobile Untuk Pemesanan Tiket Wisata Berbasis Flutter	(Unila, 2022)	Mengembangkan aplikasi mobile pemesanan tiket wisata yang mudah, efisien, dan aman dengan verifikasi tiket via QR code.	Aplikasi berjalan dengan baik, mempermudah pemesanan dan meningkatkan keamanan. Metode: <i>Prototype</i>	Belum dijelaskan secara rinci tentang performa, skalabilitas, atau beban sistem jika pengguna banyak. Bisa ditambah studi penggunaan nyata (user test dengan volume).	Penelitian ini mirip secara teknis (Flutter), namun fokus pada tiket wisata, sedangkan penelitian ini fokus pada pemesanan makanan dengan fitur transaksi dan riwayat pesanan berbasis Firebase.
8	Perancangan dan Pengembangan Aplikasi E-Commerce untuk Pemesanan Buket Bunga Berbasis Flutter: Studi Kasus Toko Bunga Kisah Kita	Acep Sandi Mutia dkk. — 2023	Membuat aplikasi e-commerce berbasis Flutter agar pelanggan dapat memilih dan memesan buket bunga secara online dengan mudah dan efisien	Aplikasi berhasil dibuat sesuai fungsinya, memudahkan pelanggan memilih dan memesan bunga. Metode: <i>Prototype</i>	Kelemahan terkait mungkin pada kustomisasi buket, manajemen stok/bunga segar, dan integrasi logistik-pengiriman; juga cakupan pengguna uji coba bisa diperluas	Penelitian ini memiliki kemiripan pada model e-commerce, namun fokus produk berbeda (bunga). Penelitian ini berfokus pada kuliner dengan sistem pesanan dan riwayat transaksi menggunakan Firebase.

No	Judul	Peneliti, dan Tahun	Tujuan Penelitian	Hasil dan Metode Yang Digunakan	Saran atau Kelemahan	Kesenjangan Penelitian
9	Perancangan dan Pembuatan Aplikasi Point of Sales UMKM berbasis Mobile menggunakan Framework Flutter (Studi Kasus: PW Café)	Ardhya Khrisna Chandra, Bayu Rahayudi, Welly Purnomo — 2022	Membantu digitalisasi operasional UMKM (PW Café) agar proses pembayaran dan pelaporan penjualan menjadi digital untuk meningkatkan efisiensi waktu operasional	Aplikasi berhasil meningkatkan efisiensi transaksi penjualan. Metode: <i>Prototype</i>	Keterbatasan: mungkin kurang fitur-layanan pelanggan, integrasi pembayaran atau keamanan mendalam, juga tidak disebut pengujian pada pengguna sangat banyak	Penelitian ini juga memakai Prototype, namun fokus pada POS internal toko, sedangkan penelitian ini pada pemesanan makanan online dengan interaksi langsung pengguna dan admin.

No	Judul	Peneliti, dan Tahun	Tujuan Penelitian	Hasil dan Metode Yang Digunakan	Saran atau Kelemahan	Kesenjangan Penelitian
10	Implementasi Mobile Apps Untuk Toko Highpom Menggunakan Framework Flutter	Aldo Mahendra dkk. — 2022	Membuat aplikasi toko online untuk Toko Highpom dengan Flutter agar pelanggan bisa melakukan pemesanan produk secara online, dan pengelola toko bisa mengupdate produk dari admin, menggunakan payment gateway midtrans	Aplikasi berhasil dikembangkan dengan UI/UX menarik dan fitur pembayaran online. Metode: <i>Prototype</i>	Saran pengembangan: menambahkan fitur pelacakan pengiriman, log aktivitas pengiriman barang untuk transparansi pengguna; juga pengujian lebih dalam terhadap skala pengguna besar	Penelitian ini berfokus pada toko online produk umum, sedangkan penelitian ini pada aplikasi kuliner dengan integrasi Firebase untuk pemesanan dan riwayat transaksi.