

BAB III

METODE PELAKSANAAN MBKM KEWIRAUSAHAAN

3.1 Produksi

Produksi aplikasi U-Teen merupakan inti dari proses pengembangan bisnis berbasis teknologi digital yang bertujuan untuk menyelesaikan berbagai permasalahan operasional kantin Universitas Multimedia Nusantara (UMN). Sebagai sebuah solusi digital, produksi aplikasi ini tidak hanya melibatkan aspek teknis pemrograman, tetapi juga melibatkan proses perancangan kebutuhan sistem, pengumpulan data pengguna, perencanaan arsitektur sistem, serta pengujian berkelanjutan untuk menjamin kualitas dan kinerja layanan yang diberikan.

Tahapan produksi dimulai dari proses identifikasi permasalahan yang terjadi di lingkungan kantin UMN, seperti antrean panjang saat jam makan siang, keterbatasan waktu istirahat mahasiswa, hingga sistem pembayaran manual yang kurang efisien. Untuk memastikan solusi yang dikembangkan benar-benar menjawab kebutuhan, tim pengembang melakukan observasi langsung di lapangan, wawancara informal dengan mahasiswa, tenant, dan pengelola kantin (Libro), serta mengkaji sistem yang telah berjalan.

Melalui proses tersebut, disusunlah desain dan arsitektur sistem aplikasi U-Teen yang mencakup fitur pemesanan makanan secara daring, sistem pre-order dan notifikasi pesanan, pembayaran digital terintegrasi, serta sistem ulasan dan rating tenant. Seluruh tahapan dalam produksi dilakukan secara kolaboratif oleh tim pengembang yang terdiri dari mahasiswa Program Studi Sistem Informasi UMN dengan pembagian peran yang jelas sesuai kompetensi.

3.1.1 Perancangan

Tahap awal produksi dimulai dengan kegiatan perancangan sistem, yaitu proses analisis kebutuhan dan perumusan solusi digital berdasarkan data lapangan. Tim U-Teen melakukan observasi langsung di kantin kampus UMN dan mendata hambatan seperti:

2. Antrean panjang pada jam makan siang,
3. Kurangnya sistem menu terpusat,
4. Keterbatasan metode pembayaran saat terjadi gangguan sistem.

Riset ini menghasilkan daftar kebutuhan yang dipecah menjadi dua kategori:

5. Kebutuhan pengguna (mahasiswa, dosen, staf): kemudahan akses menu, sistem pre-order, fleksibilitas pembayaran.
6. Kebutuhan tenant: pengelolaan menu digital, sistem notifikasi pesanan, pencatatan transaksi otomatis.

Perancangan kemudian dilanjutkan ke penyusunan arsitektur sistem, meliputi:

7. Pemilihan arsitektur Model-View-ViewModel (MVVM) untuk menjaga keterpisahan antara logika aplikasi dan antarmuka pengguna.
8. Penentuan entitas utama pada sistem basis data (users, tenants, menus, orders, transactions, reviews).
9. Pemetaan alur data antar bagian: front-end, back-end, database.

Desain UI/UX dibuat menggunakan Figma oleh CPO, dengan alur visual yang konsisten, ramah pengguna, dan menyesuaikan kebutuhan pengguna kampus yang terbiasa dengan antarmuka digital.

3.1.2 Pembangunan

Tahapan pembangunan merupakan fase penting dalam proses produksi aplikasi U-Teen, di mana semua hasil perancangan, mulai dari struktur sistem hingga antarmuka pengguna, diimplementasikan secara teknis menjadi sebuah produk digital yang dapat digunakan. Dalam konteks ini, pembangunan dilakukan secara modular dan bertahap, dengan dua fokus utama: pengembangan antarmuka pengguna (front-end) dan pengembangan sistem logika serta infrastruktur pendukung (back-end). Seluruh proses pembangunan dilakukan secara kolaboratif oleh tim pengembang, dengan distribusi tanggung jawab yang jelas dan integrasi hasil kerja yang terkoordinasi.

A. Pengembangan Antarmuka Pengguna (Front-End)

Antarmuka pengguna (user interface) merupakan wajah dari aplikasi U-Teen. Semua interaksi yang dilakukan oleh pengguna terjadi melalui komponen front-end. Untuk memastikan aplikasi dapat diakses oleh seluruh civitas akademika UMN baik pengguna Android maupun iOS, tim menggunakan Flutter sebagai framework pengembangan utama. Flutter dipilih karena kemampuannya untuk membangun aplikasi lintas platform dengan satu basis kode, yang secara signifikan menghemat waktu dan sumber daya dalam pengembangan serta pemeliharaan.

Bahasa pemrograman yang digunakan dalam pengembangan front-end adalah Dart, yang merupakan bahasa bawaan Flutter. Dengan pendekatan ini, tim dapat menghasilkan antarmuka yang responsif, stabil, dan konsisten di berbagai perangkat.

Seluruh antarmuka aplikasi dirancang berdasarkan prinsip responsive design, agar dapat menyesuaikan tampilan dengan ukuran layar yang berbeda. Ini penting mengingat pengguna memiliki beragam tipe perangkat mulai dari smartphone entry-level hingga perangkat flagship. Implementasi UI mengikuti desain visual yang sebelumnya dibuat dengan Figma oleh tim CPO. Setiap komponen, seperti tombol, form, navigasi, dan ikon, diprogram agar mengikuti standar tampilan modern dengan memperhatikan kemudahan akses, konsistensi, dan kenyamanan pengguna.

Struktur navigasi aplikasi disusun agar intuitif dan mudah dipahami, dimulai dari halaman login dan registrasi, menuju dashboard utama yang menampilkan berbagai menu makanan, halaman detail pemesanan, sistem pembayaran dengan tampilan QR Code, status pesanan secara real-time, hingga histori transaksi dan fitur ulasan tenant.

Tim juga memastikan transisi antarhalaman berjalan lancar melalui penggunaan animasi ringan, serta memanfaatkan elemen visual seperti warna, ikonografi, dan tipografi yang sesuai dengan karakteristik pengguna utama, yaitu mahasiswa.

B. Pengembangan Sistem Backend

Sementara antarmuka menjadi sisi yang dilihat pengguna, sistem backend merupakan jantung dari logika aplikasi. Backend bertanggung jawab untuk mengatur bagaimana data dikelola, diproses, disimpan, dan ditransmisikan antar pihak dalam aplikasi. Dalam pembangunan backend U-Teen, teknologi utama yang digunakan adalah Firebase, layanan Backend-as-a-Service (BaaS) milik Google yang menyediakan berbagai fitur pendukung pengembangan aplikasi modern.

Komponen utama yang dibangun dalam sistem backend meliputi:

1. **Firestore:** digunakan untuk menangani proses login dan pendaftaran pengguna. Setiap pengguna yang masuk akan diberikan UID (User ID) unik yang kemudian dikaitkan dengan data mereka di database.
2. **Cloud Firestore:** digunakan sebagai basis data utama aplikasi. Firestore dipilih karena bersifat NoSQL, mudah diskalakan, dan mendukung sinkronisasi data secara real-time, sehingga cocok untuk sistem pemesanan yang memerlukan pembaruan status seketika.
3. **Cloud Functions:** digunakan untuk menjalankan logika otomatisasi di server. Contohnya, ketika status pesanan berubah, sistem akan otomatis mengirimkan notifikasi kepada pengguna tanpa perlu tindakan manual.
4. **Storage:** digunakan untuk menyimpan file seperti gambar menu makanan dan bukti pembayaran. File disimpan secara efisien dan ditautkan ke metadata pengguna melalui URL yang aman.

Dalam mengelola alur backend, bahasa pemrograman Dart kembali digunakan, karena Flutter SDK menyediakan integrasi langsung untuk berkomunikasi dengan Firebase. Dengan pendekatan ini, semua fitur dapat dikembangkan dalam satu ekosistem bahasa, sehingga mempercepat

debugging dan pengujian. Untuk aspek keamanan, penulis menyusun dan mengimplementasikan Firebase Security Rules guna mengatur hak akses data. Misalnya, tenant hanya dapat mengakses pesanan yang dibuat untuk toko mereka, dan pengguna hanya dapat melihat historinya sendiri. Pengaturan ini sangat krusial agar tidak terjadi kebocoran data antar peran.

Selama tahap pembangunan, pendekatan iteratif dan agile diterapkan. Setiap fitur dikembangkan dalam siklus singkat: dirancang, dibangun, diuji secara internal, lalu diintegrasikan ke sistem utama. Hal ini memastikan jika ada kesalahan (bug) atau kekurangan fitur, dapat segera diperbaiki sebelum berdampak ke fitur lain. Integrasi antara front-end dan back-end dilakukan secara menyeluruh melalui mekanisme asynchronous call menggunakan Future dan StreamBuilder dalam Flutter. Setiap permintaan data dari pengguna dikirim ke database dan ditangani secara real-time, termasuk notifikasi perubahan status pesanan, riwayat pembelian, atau pembaruan daftar menu tenant. Proses pembangunan ini juga melibatkan dokumentasi menyeluruh atas struktur kode, alur data, dan struktur database agar dapat digunakan kembali dan dikembangkan lebih lanjut oleh pihak lain, baik di dalam kampus maupun oleh institusi pendidikan yang ingin mereplikasi sistem.

3.1.3 Database

Dalam pengembangan U-Teen, Firebase digunakan sebagai platform utama untuk pengelolaan data backend, khususnya dengan memanfaatkan Cloud Firestore sebagai basis data NoSQL. Firestore memungkinkan aplikasi menangani data dalam bentuk dokumen dan koleksi yang fleksibel, cepat, dan mudah diakses secara real-time, sehingga sangat cocok untuk aplikasi mobile modern seperti U-Teen.

Struktur database Firestore pada U-Teen dirancang sebagai berikut:

1. users: Koleksi yang menyimpan dokumen setiap pengguna. Masing-masing dokumen memiliki struktur field seperti:
 - name: Nama lengkap pengguna
 - nim: Nomor induk mahasiswa

- email: Alamat email pengguna (menggunakan domain UMN)
 - phoneNumber: Nomor telepon
 - angkatan: Tahun angkatan mahasiswa
 - prodi: Program studi pengguna
 - userType: Peran pengguna dalam aplikasi, seperti "customer", "tenant", atau "admin"
2. cart: Koleksi yang menyimpan daftar makanan yang ditambahkan pengguna ke keranjang belanja. Koleksi ini berada di dalam subkoleksi setiap pengguna.
 3. products: Menyimpan daftar menu makanan dari seluruh tenant, termasuk informasi seperti nama produk, harga, kategori, deskripsi, dan ID tenant.
 4. orders: Menyimpan data pemesanan makanan. Field umumnya mencakup ID pengguna, waktu pemesanan, item yang dipesan, total harga, metode pembayaran, dan status pesanan.
 5. payments (opsional, jika disimpan terpisah): Menyimpan informasi status dan metode pembayaran dari setiap order.
 6. favorites: Koleksi untuk menyimpan makanan favorit tiap pengguna.
 7. notifications: Koleksi berisi notifikasi pesanan atau promosi untuk pengguna, dikirim melalui Firebase Cloud Messaging (FCM).
 8. popular_cuisines dan search_history: Digunakan untuk mendukung fitur eksplorasi menu dan rekomendasi berdasarkan interaksi pengguna.

Firestore Authentication digunakan untuk login menggunakan akun @student.umn.ac.id, sementara otorisasi pengguna ditentukan berdasarkan field userType pada koleksi users. Cloud Functions digunakan untuk logika backend seperti pemrosesan status pesanan otomatis dan pengiriman notifikasi. Sementara itu, Firebase Cloud Messaging (FCM) memungkinkan

sistem memberikan notifikasi real-time kepada pengguna ketika pesanan sedang diproses atau siap diambil.

Keunggulan Firebase yang Dimanfaatkan:

1. Skalabilitas tanpa pengelolaan server manual
2. Realtime data sync untuk respons cepat terhadap perubahan status pesanan
3. Keamanan data melalui Firebase Security Rules
4. Integrasi langsung dengan Flutter SDK untuk sinkronisasi frontend-backend

Dengan penggunaan Firebase, aplikasi U-Teen dapat dikembangkan dengan cepat, aman, dan andal tanpa infrastruktur server yang kompleks. Struktur data berbasis dokumen juga membuat pengelolaan pengguna, menu, dan transaksi menjadi lebih fleksibel.

3.1.4 Pengujian dan Evaluasi

Tahap akhir produksi adalah pengujian dan evaluasi sistem yang bertujuan memastikan aplikasi berjalan sesuai harapan dan siap diimplementasikan di lingkungan kampus.

A. Jenis Pengujian

1. Unit Testing: Pengujian fungsi-fungsi individual seperti pemesanan, login, dan transaksi.
2. Integration Testing: Menguji alur login-pesan-bayar dalam satu rangkaian skenario.
3. Emulator Testing: Simulasi di berbagai ukuran dan jenis perangkat.
4. Real Device Testing: Uji coba aplikasi di perangkat fisik Android dan iOS.

B. Uji Lapangan (User Acceptance Test)

1. Aplikasi diuji langsung oleh tenant dan mahasiswa dalam kondisi operasional nyata.

2. Pengujian dilakukan di lingkungan kantin dengan skenario antrean dan waktu makan siang.

C. Hasil Evaluasi

1. Respons pengguna positif, terutama terhadap fitur pre-order dan notifikasi pesanan.
2. Notifikasi real-time berfungsi dengan baik, namun perlu optimalisasi gambar agar lebih ringan.
3. Tenant merasa terbantu dengan sistem pemantauan transaksi otomatis dan dashboard menu digital.
4. Ditemukan bug minor seperti delay tampilan status pemesanan dan input data rating, yang langsung diperbaiki melalui hotfix.

D. Perbaikan Sistem

1. Penambahan fitur loading animation dan toast feedback.
2. Optimalisasi query database untuk menghindari fetch delay.
3. Perbaikan desain UI berdasarkan masukan pengguna.

3.2 Penetapan Harga

Strategi penetapan harga merupakan salah satu aspek penting dalam perencanaan dan pengembangan usaha berbasis teknologi, khususnya pada tahap awal ketika model bisnis sedang diuji dan divalidasi di lapangan. Dalam konteks aplikasi U-Teen, strategi harga dirancang dengan pendekatan berbasis nilai (value-based pricing), di mana harga yang dikenakan disesuaikan dengan manfaat langsung yang dirasakan oleh pengguna serta efisiensi yang diberikan kepada mitra usaha, yaitu tenant kantin kampus.

Model harga ini tidak hanya mempertimbangkan aspek keuntungan, tetapi juga mendukung prinsip inklusivitas digital, keberlanjutan operasional, serta adopsi teknologi yang tinggi dari sisi pengguna dan mitra. Penetapan harga yang terlalu tinggi berisiko menghambat pertumbuhan adopsi, sementara harga yang terlalu rendah dapat mengancam keberlangsungan layanan. Oleh karena itu, keseimbangan

antara manfaat dan biaya menjadi prinsip utama dalam strategi penetapan harga U-Teen.

3.2.1 Berdasarkan Nilai yang Diberikan

Sebagai solusi digital berbasis kampus, U-Teen mengadopsi strategi gratis untuk pengguna akhir, yaitu mahasiswa, dosen, dan staf UMN. Strategi ini diterapkan untuk memaksimalkan jumlah pengguna dan menurunkan hambatan adopsi sistem, khususnya di fase awal implementasi. Tidak ada biaya langganan, biaya pemesanan, atau biaya tambahan yang dibebankan kepada pengguna aplikasi.

Sebaliknya, sumber pendapatan utama berasal dari tenant yang mendapatkan manfaat operasional langsung dari sistem ini. Tenant tidak hanya mendapatkan pemesanan yang lebih tertata, tetapi juga pengelolaan menu yang lebih rapi, pencatatan transaksi otomatis, serta exposure digital yang lebih merata di antara tenant lain.

Dengan pendekatan ini, U-Teen menempatkan dirinya sebagai mitra operasional tenant yang turut membantu meningkatkan efisiensi, akurasi layanan, dan kepuasan pelanggan. Maka dari itu, biaya yang dikenakan bersifat berbasis transaksi (per order) dan bukan biaya tetap, sehingga tetap adil untuk tenant dengan volume transaksi kecil maupun besar.

3.2.2 Sistem Bagi Hasil

Model bisnis utama yang digunakan oleh U-Teen adalah sistem bagi hasil (revenue sharing) dengan tenant. Setiap transaksi yang terjadi melalui aplikasi akan menghasilkan potongan komisi sebesar 2% dari total nilai transaksi. Berikut ilustrasi pembagiannya:

1. Tenant menerima 75% dari nilai transaksi secara langsung sebagai pendapatan bersih.
2. 25% sisanya dibagi antara U-Teen (sebagai pengembang dan pengelola aplikasi) dan pihak pengelola kantin (Libro) sebagai penyedia sistem operasional fisik.

Skema ini dirancang untuk menciptakan kemitraan yang sehat antara ketiga pihak:

1. Tenant tetap memperoleh mayoritas pendapatan.
2. Pihak pengelola mendapat insentif untuk mendukung implementasi teknologi.
3. U-Teen memiliki pemasukan yang proporsional untuk mendukung operasional dan pengembangan sistem.

3.2.3 Komisi dari Sistem Pembayaran

Dalam mendukung transaksi digital, U-Teen bermitra dengan penyedia layanan pembayaran seperti GoPay dan OVO, melalui integrasi sistem dengan pihak ketiga (seperti Midtrans). Platform-platform ini mengenakan biaya layanan transaksi sebagai berikut:

1. GoPay: 1,5% per transaksi,
2. OVO: 2,9% per transaksi.

Biaya tersebut sebagian besar masuk ke penyedia layanan pembayaran, namun U-Teen mendapatkan komisi teknologi tambahan dari kerja sama ini, tergantung pada volume transaksi bulanan.

Pengguna tidak dibebankan tambahan biaya dari transaksi ini karena sistem telah dioptimalkan agar potongan terjadi di backend melalui pengaturan merchant fee. Dengan demikian, pengguna tetap merasakan kenyamanan dan transparansi harga tanpa biaya tersembunyi.

3.2.4 Strategi Harga yang Berkelanjutan

Salah satu tantangan utama dalam pengembangan startup digital di sektor pendidikan adalah menjaga keberlanjutan operasional. Oleh karena itu, strategi harga U-Teen dirancang tidak hanya untuk menutup biaya awal, tetapi juga untuk menopang ekspansi dan pemeliharaan sistem jangka panjang.

Selama tahap awal program MBKM, seluruh biaya server, promosi, dan pengembangan ditanggung oleh kampus (UMN) dan inkubator Skystar

Ventures. Namun, untuk tahap pasca-program, pendapatan dari komisi transaksi akan digunakan untuk:

1. Membiayai biaya operasional server dan keamanan data,
2. Menyediakan layanan pemeliharaan sistem dan perbaikan bug,
3. Mengembangkan fitur tambahan seperti loyalty program, integrasi POS (Point of Sale), dan pelaporan keuangan tenant,
4. Mendanai promosi dan kampanye adopsi pengguna di masa depan.

Dengan struktur harga yang ringan namun berkelanjutan ini, U-Teen mampu membangun model usaha yang tidak membebani pihak manapun secara tidak proporsional.

3.2.5 Potensi Ekspansi Pasar

Salah satu kekuatan utama dari struktur harga U-Teen adalah fleksibilitasnya untuk direplikasi ke berbagai kampus lain di Indonesia. Karena sistem ini tidak membebankan biaya tetap dan hanya mengenakan komisi dari transaksi yang terjadi, maka:

1. Kampus lain dapat mengadopsi sistem ini tanpa biaya besar di awal,
2. Tenant di kampus baru tidak perlu khawatir membayar biaya tetap jika belum memiliki banyak pelanggan,
3. U-Teen tetap memperoleh pendapatan sesuai dengan performa penggunaan aplikasi.

Model ini menciptakan ekosistem yang insentif-aligned, di mana semua pihak akan terdorong untuk meningkatkan adopsi sistem demi mendapatkan keuntungan bersama

3.3 Promosi Target Pasar

3.3.1 Target Pasar

Target pasar utama dari aplikasi U-Teen adalah seluruh civitas akademika Universitas Multimedia Nusantara (UMN), meliputi mahasiswa, dosen, dan staf kampus yang secara rutin menggunakan layanan kantin untuk memenuhi kebutuhan konsumsi harian. Selain pengguna akhir, kelompok pasar strategis lainnya adalah tenant kantin yang berperan sebagai

mitra operasional dalam sistem digital ini.

Pemilihan target pasar ini didasarkan pada dua indikator utama. Pertama, tingginya volume transaksi harian di lingkungan kantin UMN, yang mencapai ±Rp60.000.000 pada hari kerja dan ±Rp10.000.000 pada hari Sabtu. Kedua, kebutuhan nyata terhadap efisiensi sistem, terutama dalam hal pemesanan makanan, pengelolaan antrean, dan fleksibilitas pembayaran digital

Dengan jumlah mahasiswa aktif dan staf yang signifikan, serta budaya penggunaan teknologi yang sudah tinggi di lingkungan kampus, potensi adopsi terhadap aplikasi ini sangat besar. Oleh karena itu, strategi penetrasi pasar difokuskan pada penyediaan solusi digital yang tidak hanya praktis, tetapi juga memiliki nilai tambah operasional bagi kedua sisi pengguna dan tenant.

3.3.2 Metode Promosi

Untuk mencapai penetrasi pasar yang maksimal dan mendorong adopsi sistem secara luas, tim U-Teen mengembangkan pendekatan promosi ganda: memanfaatkan saluran internal kampus (promosi tertarget) dan pendekatan berbasis perilaku pengguna (promosi sosial dan interaktif).

A. Promosi Langsung Melalui Kanal Resmi Kampus

1. Email Resmi UMN:

Tim bekerja sama dengan pihak kampus untuk melakukan email blast ke seluruh mahasiswa, dosen, dan staf menggunakan sistem email resmi. Konten email mencakup informasi fitur, manfaat, instruksi pengunduhan, serta link ke aplikasi.

2. Aplikasi Union (Sistem Internal Kampus):

Sebagai platform yang umum digunakan oleh mahasiswa, promosi melalui aplikasi Union memungkinkan eksposur tinggi. Informasi ditampilkan dalam bentuk banner dan pop-up saat pengguna mengakses Union.

3. Dukungan Dosen dan Staf Akademik:

Pihak fakultas dan pengajar yang telah mengetahui nilai dari aplikasi didorong untuk ikut mempromosikan penggunaan U-Teen di kelas atau aktivitas kampus.

B. Promosi di Lingkungan Kantin

1. Tenant Engagement:

Tenant yang sudah bergabung dalam sistem U-Teen didorong untuk mempromosikan aplikasi secara langsung kepada pelanggan. Contohnya, kasir tenant menyarankan pengguna untuk melakukan pemesanan melalui aplikasi dengan pertanyaan seperti, “Sudah coba pesan lewat U-Teen belum?”

2. Banner dan Standing Poster:

Materi promosi visual seperti poster, standing banner, dan QR Code dipasang di titik-titik strategis di area kantin untuk menarik perhatian mahasiswa yang lalu lalang.

3. Demo Langsung dan Booth Promosi:

Tim U-Teen mengadakan sesi demonstrasi langsung pada jam sibuk di kantin untuk memperlihatkan cara penggunaan aplikasi dan manfaatnya dalam waktu nyata.

C. Promosi Berbasis Komunitas dan Rekomendasi

1. Efek Rekomendasi Teman:

Pengalaman positif pengguna pertama (early adopter) dimanfaatkan untuk menciptakan efek viral alami, di mana pengguna akan merekomendasikan aplikasi kepada teman-temannya. Strategi ini terbukti efektif karena didasarkan pada kepercayaan dan pengalaman langsung.

2. Insentif Awal dan Promo Eksklusif:

Pengguna yang pertama kali mencoba aplikasi diberikan diskon atau cashback dalam bentuk saldo GoPay/OVO. Sementara tenant yang aktif mempromosikan U-Teen diberikan akses fitur premium terlebih dahulu secara gratis.

3.3.3 Potensi Pertumbuhan dan Replikasi Pasar

Meskipun saat ini promosi difokuskan di lingkungan UMN, strategi promosi jangka panjang telah disiapkan untuk memungkinkan replikasi sistem ke kampus-kampus lain. Karena arsitektur aplikasi bersifat fleksibel dan scalable, strategi promosi dapat diadaptasi sesuai dengan ekosistem kampus masing-masing.

Strategi promosi untuk ekspansi mencakup:

1. Kerja sama dengan biro kemahasiswaan kampus lain,
2. Kolaborasi dengan perusahaan penyedia makanan (catering universitas),
3. Partisipasi dalam pameran inovasi mahasiswa dan kompetisi startup pendidikan.

Dengan menggabungkan pendekatan digital-first, komunitas, dan kemitraan institusional, promosi U-Teen dirancang untuk menciptakan ekosistem adopsi yang berkelanjutan dan progresif di sektor pendidikan tinggi.

3.4 Tahapan Pekerjaan yang Dilakukan Dalam MBKM Kewirausahaan

Tabel 3.1 Detail Pekerjaan yang Dilakukan Dalam MBKM Cluster Kewirausahaan

No.	Minggu	Proyek	Keterangan
1	1-2	Perancangan Ide Bisnis	Melakukan brainstorming untuk perancangan ide bisnis dan aplikasi untuk MBKM kewirausahaan
2	2-4	Perencanaan Pengembangan Database untuk aplikasi	Melakukan riset mengenai bahasa pemrograman dan aplikasi database mana yang cocok untuk digunakan dalam pengembangan ide bisnis
3	5-8	Perancangan Database untuk Customer	Melakukan perancangan database untuk customer pada aplikasi
4	8-10	Perancangan Database untuk Tenant	Melakukan perancangan database untuk tenant pada aplikasi
5	10-14	Optimalisasi Database dan Fitur pada Aplikasi	Melakukan penghubungan database terhadap aplikasi yang sudah dibuat dan memastikan bahwa semua fitur dapat berjalan dengan

			lancar dan pengambilan data dari database berjalan dengan lancar
--	--	--	--

3.5 Uraian Pelaksanaan Kerja Dalam MBKM Kewirausahaan

3.5.1 Proses Pelaksanaan

Uraian berfokus pada beberapa bagian pekerjaan yang penulis lakukan dalam MBKM Cluster Kewirausahaan. Jumlah bagian dikonsultasikan dengan dosen pembimbing. Penjelasan pada bagian ini harus rinci dan menggambarkan apa yang penulis kerjakan. Foto-foto hasil dan proses pekerjaan yang penulis lakukan dapat ditampilkan pula di bagian ini. Jumlah minimal proses yang dijabarkan adalah 5 proyek atau 5 karya, meliputi proses perancangan dari awal hingga akhir.

A. Proyek Database U-Teen

A.1 Firebase

Firebase merupakan platform Backend-as-a-Service (BaaS) yang dikembangkan oleh Google dan dirancang untuk mendukung pengembangan aplikasi web dan mobile secara modern. Firebase menyatukan berbagai layanan penting seperti Realtime Database, Cloud Firestore, Authentication, dan Cloud Functions dalam satu ekosistem terpadu yang dapat diakses dengan mudah oleh pengembang. Arsitekturnya yang berbasis cloud memungkinkan pengelolaan data secara real-time, integrasi lintas platform, serta deployment sistem backend tanpa kebutuhan infrastruktur server fisik.

Pada proyek U-Teen, Firebase dipilih sebagai fondasi utama dalam perancangan dan pengelolaan sistem basis data. Salah satu fitur utama yang dimanfaatkan adalah Firebase Realtime Database, yaitu sistem NoSQL berbasis JSON yang memungkinkan data diperbarui secara instan di seluruh klien yang terhubung. Fitur ini sangat ideal untuk mendukung skenario seperti pemesanan makanan secara langsung, pelacakan status pesan, dan notifikasi pesanan siap ambil secara real-time. Setiap perubahan data

tersinkronisasi secara otomatis tanpa perlu intervensi tambahan dari sisi pengguna, menciptakan pengalaman interaktif yang lancar.

Selain itu, Firebase Authentication digunakan untuk menangani sistem login dan manajemen akun pengguna. Pengguna dapat masuk menggunakan akun email, akun Google, atau melalui integrasi sistem Single Sign-On (SSO) kampus. Hal ini memastikan keamanan dan kenyamanan dalam proses otorisasi pengguna aplikasi U-Teen.

Firebase juga dilengkapi Cloud Functions, layanan serverless yang memungkinkan eksekusi logika bisnis berbasis event. Misalnya, saat pengguna menyelesaikan pembayaran, sistem dapat secara otomatis memicu fungsi untuk mengubah status pesanan, mengirim notifikasi ke tenant, atau mencatat transaksi ke log aktivitas. Mekanisme ini meningkatkan efisiensi sistem sekaligus menjaga arsitektur backend tetap ringan dan terdesentralisasi.

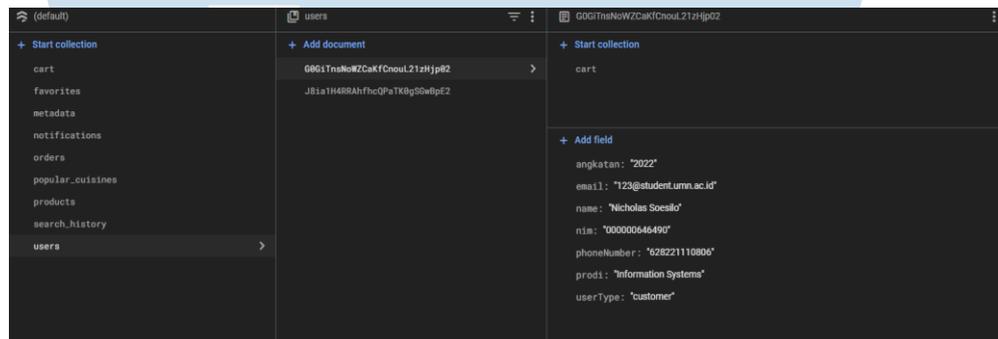
Alasan utama dipilihnya Firebase adalah kompatibilitas dan kemudahan integrasinya dengan Flutter framework utama dalam pengembangan antarmuka U-Teen. Firebase menyediakan SDK resmi untuk Flutter, yang memungkinkan penghubungan langsung antara aplikasi frontend dan layanan backend tanpa konfigurasi kompleks. Hal ini mempercepat proses pengembangan, mendukung modularitas arsitektur, dan memastikan kestabilan koneksi antar komponen.

Dari sisi keamanan, Firebase menyediakan sistem Firebase Security Rules yang memungkinkan untuk mendefinisikan hak akses secara granular terhadap setiap entitas data dalam database. Misalnya, hanya tenant tertentu yang dapat mengakses data transaksi mereka, atau hanya pengguna terverifikasi yang dapat membuat pesanan.

Dalam konteks proyek U-Teen, seluruh struktur database dirancang menggunakan pendekatan document-oriented. Setiap entitas seperti pengguna, pesanan, tenant, dan riwayat transaksi disusun dalam koleksi dan dokumen yang saling terhubung. Pendekatan ini memudahkan proses pembacaan data secara terstruktur dan efisien.

Firestore dipilih bukan hanya karena kelengkapan fiturnya, tetapi juga karena kemampuannya dalam mendukung pertumbuhan aplikasi berskala besar di masa depan. Dengan fitur seperti automatic scaling, distributed database, dan monitoring bawaan, Firestore menjamin performa sistem tetap stabil seiring bertambahnya pengguna.

Secara keseluruhan, penggunaan Firestore dalam proyek U-Teen memungkinkan pengembangan sistem yang ringan, real-time, fleksibel, dan terintegrasi penuh. Firestore berperan tidak hanya sebagai fondasi penyimpanan data, tetapi juga sebagai jembatan utama antara logika backend, antarmuka pengguna, dan kebutuhan bisnis digital kampus modern.



Gambar 3.1 Struktur table users database

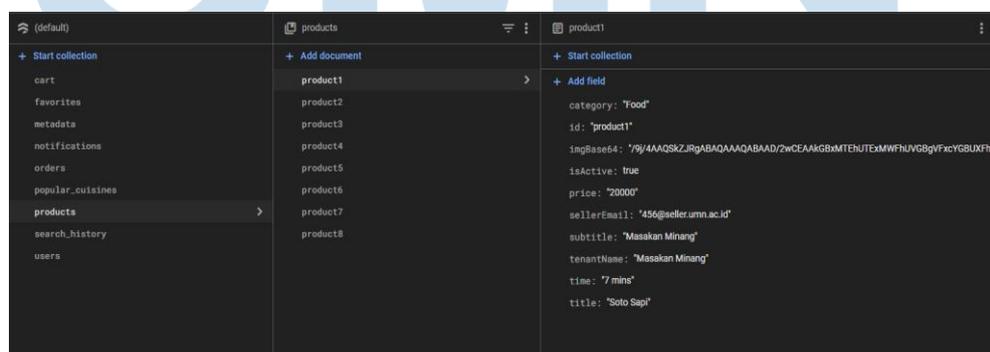
Pada Gambar 3.1 dapat dilihat bahwa koleksi users merupakan komponen fundamental dalam sistem backend aplikasi U-Teen karena merepresentasikan seluruh entitas pengguna yang terlibat dalam aplikasi. Koleksi ini mencakup data untuk berbagai jenis pengguna seperti mahasiswa (customer), tenant (penjual makanan), serta admin (pengelola sistem). Setiap dokumen di dalam koleksi ini memiliki field identitas utama seperti name, nim, email, angkatan, prodi, dan userType. Field userType berfungsi sebagai acuan sistem untuk memberikan antarmuka dan hak akses yang sesuai dengan peran pengguna.

Struktur ini juga dirancang untuk mengintegrasikan subkoleksi cart, yang menyimpan item yang telah dimasukkan pengguna ke dalam keranjang belanja sebelum melakukan checkout. Subkoleksi cart disusun dalam bentuk array dari produk yang dipilih, termasuk jumlah dan detail harga, sehingga

dapat diproses saat transaksi dilakukan. Pendekatan ini memberikan fleksibilitas dalam pemrosesan data dan mendukung modularitas sistem. Dengan adanya subkoleksi ini, interaksi pengguna terhadap produk dapat direkam secara independen tanpa memengaruhi data utama profil pengguna.

Koleksi users ini juga menjadi dasar bagi sistem autentikasi dan personalisasi layanan. Firebase Authentication akan menghubungkan setiap akun pengguna dengan data di koleksi ini, memungkinkan aplikasi untuk menampilkan halaman yang sesuai berdasarkan peran pengguna. Misalnya, tenant akan diarahkan ke halaman manajemen produk, sementara mahasiswa diarahkan ke halaman eksplorasi menu dan pemesanan makanan. Integrasi ini juga mendukung implementasi fitur login otomatis dengan akun kampus melalui sistem Single Sign-On (SSO).

Keamanan pada struktur ini dijaga menggunakan Firebase Security Rules, di mana pengguna hanya dapat mengakses dan memodifikasi dokumen milik mereka sendiri. Hal ini sangat penting dalam menjaga integritas data dan privasi, terutama pada sistem yang melibatkan data pribadi mahasiswa dan transaksi digital. Dengan desain yang skalabel dan fleksibel, struktur users pada Gambar 3.1 memungkinkan pengembangan fitur lanjutan seperti riwayat login, preferensi pengguna, serta notifikasi individual yang dapat mendukung pengalaman pengguna yang lebih personal dan aman.



Gambar 3.2 Struktur Table Database Products

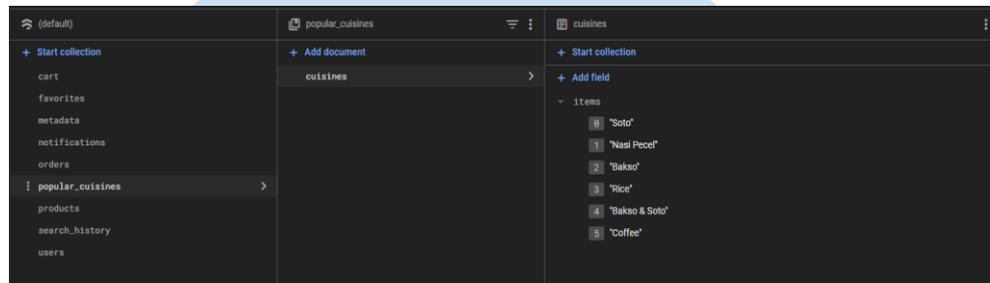
Gambar 3.2 memperlihatkan struktur koleksi products, yang berfungsi sebagai pusat data untuk seluruh menu makanan dan minuman yang dijual oleh tenant di kantin UMN melalui aplikasi U-Teen. Setiap dokumen dalam koleksi ini menyimpan informasi lengkap mengenai satu produk, seperti productName, description, price, category, imageUrl, dan tenantId. Struktur ini memungkinkan aplikasi untuk menampilkan daftar produk berdasarkan tenant, kategori makanan, atau hasil pencarian tertentu yang dilakukan pengguna.

Field tenantId sangat penting dalam koleksi ini karena menjadi relasi antara produk dan pemiliknya, sehingga sistem dapat secara otomatis menampilkan hanya produk yang dimiliki oleh tenant tertentu. Selain itu, penggunaan imageUrl juga memungkinkan visualisasi makanan secara langsung pada antarmuka aplikasi. Gambar-gambar menu ini tidak hanya mempercantik tampilan tetapi juga memainkan peran besar dalam meningkatkan minat beli pengguna. Visualisasi ini terhubung dengan Firebase Storage, tempat gambar produk diunggah dan diakses secara real-time.

Struktur products juga mendukung pemrosesan dinamis secara real-time. Apabila tenant mengubah harga atau detail produk, data akan langsung diperbarui di sisi pengguna tanpa memerlukan refresh manual berkat keunggulan Firebase Realtime Database dan Firestore. Hal ini sangat membantu dalam menjaga konsistensi data dan efisiensi operasional tenant, terutama dalam skenario di mana produk sering mengalami perubahan, seperti menu harian atau penyesuaian harga.

Selain itu, koleksi ini dapat diperluas untuk menyertakan informasi tambahan seperti status promosi (misalnya diskon), ketersediaan stok, dan tag khusus seperti “spicy” atau “vegetarian”. Fitur-fitur ini memberikan fleksibilitas lebih dalam pengelolaan produk serta memudahkan pengguna dalam melakukan eksplorasi menu. Gambar 3.2 mencerminkan bagaimana

struktur data yang sederhana namun kuat ini menjadi fondasi utama sistem digitalisasi menu di lingkungan kantin modern berbasis aplikasi.

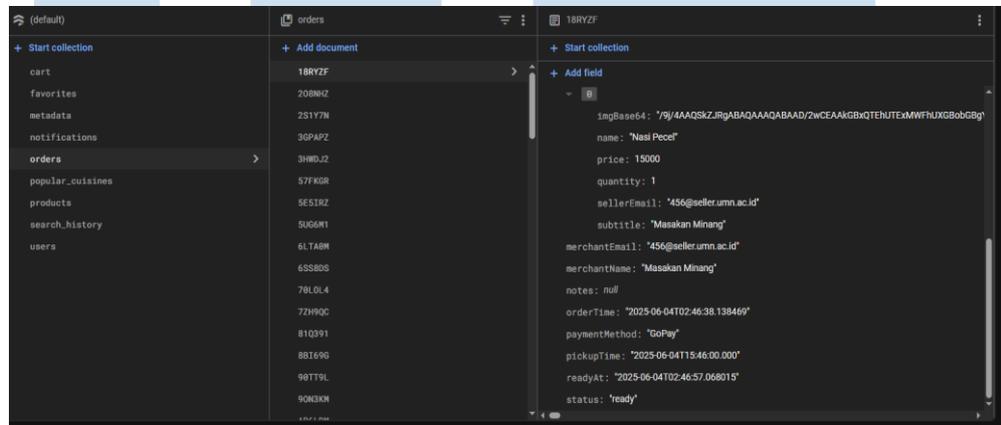


Gambar 3.3 Struktur Table Database Popular Cuisines

Gambar 3.3 menunjukkan struktur koleksi `popular_cuisines`, yang digunakan dalam sistem rekomendasi menu pada aplikasi U-Teen. Koleksi ini menyimpan data mengenai jenis-jenis makanan atau minuman yang paling populer berdasarkan aktivitas pengguna, seperti jumlah pesanan, rating tertinggi, atau frekuensi dilihat. Field-field penting yang terdapat dalam struktur ini antara lain `name`, `category`, dan `popularityScore`, di mana `popularityScore` dihitung dari kombinasi metrik interaksi pengguna terhadap produk tertentu. Data pada koleksi ini diambil dari histori transaksi pengguna dan diolah secara berkala melalui Cloud Functions atau sistem analitik Firebase. Hasilnya kemudian digunakan untuk menghasilkan rekomendasi dinamis kepada pengguna seperti “Menu Paling Diminati Hari Ini” atau “Favorit Mahasiswa Minggu Ini”. Dengan demikian, aplikasi tidak hanya menampilkan menu secara statis, tetapi juga menyesuaikan konten berdasarkan preferensi dan tren aktual di lingkungan kampus. Pendekatan ini memberikan nilai tambah dalam hal personalisasi dan meningkatkan keterlibatan pengguna.

Selain itu, struktur ini juga sangat bermanfaat bagi tenant, karena dapat digunakan sebagai indikator performa produk yang mereka jual. Tenant dapat mengetahui produk mana yang paling populer dan mempertimbangkan strategi promosi atau penyesuaian menu berdasarkan data ini. Dalam pengembangan lebih lanjut, struktur ini juga dapat menjadi dasar untuk sistem rekomendasi berbasis machine learning seperti collaborative filtering, yang

dapat menyarankan menu berdasarkan kesamaan pola pesanan antar pengguna. Dengan keberadaan `popular_cuisines` seperti yang digambarkan pada Gambar 3.3, aplikasi U-Teen dapat memberikan pengalaman eksplorasi kuliner yang lebih menarik, informatif, dan sesuai dengan kebiasaan pengguna. Sistem rekomendasi ini menjadikan aplikasi tidak hanya sebagai alat transaksi, tetapi juga sebagai platform penunjang keputusan dalam memilih makanan di lingkungan kampus secara cerdas dan berbasis data.



Gambar 3.4 Struktur Table Database Orders

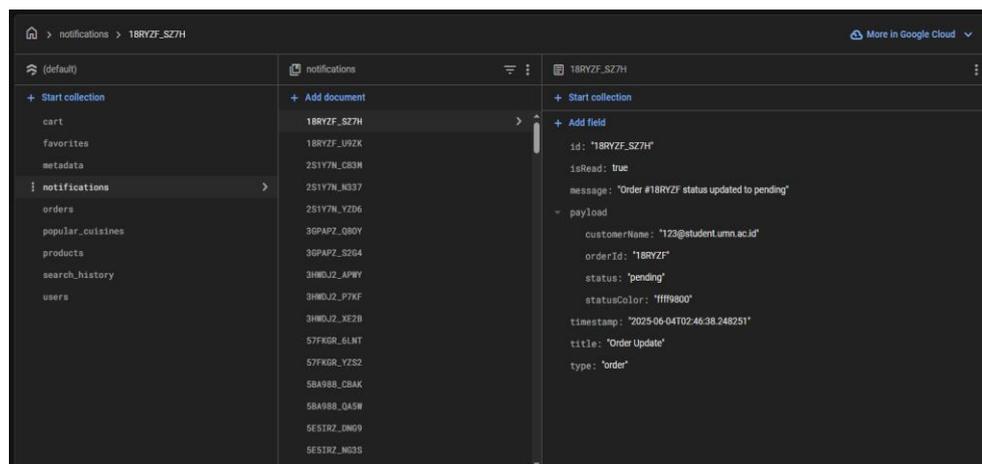
Pada Gambar 3.4 diperlihatkan struktur koleksi `orders`, yang digunakan untuk menyimpan data setiap pemesanan makanan oleh pengguna. Koleksi ini menjadi komponen kunci dalam sistem karena menangani seluruh transaksi antara customer dan tenant, termasuk status pemrosesan pesanan. Setiap dokumen berisi field seperti `orderId`, `userId`, `tenantId`, `orderItems`, `pickupTime`, `paymentMethod`, `status`, dan `timestamp`. Struktur ini memungkinkan pemesanan makanan dilakukan secara efisien dan terstruktur, bahkan untuk pesanan yang terdiri dari banyak produk sekaligus.

Field `orderItems` biasanya berbentuk array dari produk yang dipesan, lengkap dengan kuantitas dan harga masing-masing, sehingga sistem dapat menghitung total pembayaran secara otomatis. Field `pickupTime` memungkinkan pengguna memilih waktu pengambilan makanan sesuai preferensi mereka, sehingga antrean dan waktu tunggu bisa dikurangi secara

signifikan. Selain itu, field status mendukung pelacakan proses pemesanan secara real-time, seperti “dipesan”, “diproses”, “siap ambil”, hingga “selesai”.

Data dalam koleksi orders tidak hanya digunakan untuk menampilkan status pesanan, tetapi juga dapat dimanfaatkan untuk laporan transaksi oleh tenant maupun admin. Tenant dapat melihat daftar pesanan yang masuk dan menyelesaikannya sesuai urutan, sementara admin dapat memonitor aktivitas harian, volume transaksi, dan efektivitas layanan. Cloud Functions juga memungkinkan sistem mengubah status pesanan secara otomatis berdasarkan event tertentu, seperti pesanan yang telah dibayar atau waktu ambil yang sudah dekat.

Struktur ini juga dapat diperluas dengan fitur tambahan seperti penilaian pasca-transaksi, pelacakan refund, atau integrasi pembayaran digital seperti Midtrans. Dengan dukungan arsitektur Firebase yang fleksibel, koleksi orders pada Gambar 3.4 menjadi pusat kendali dari seluruh interaksi transaksi dalam aplikasi U-Teen, sekaligus menyediakan fondasi data untuk analitik dan evaluasi performa tenant.



Gambar 3.5 Struktur Table Database Notifications

Gambar 3.5 menggambarkan struktur koleksi notifications, yang berfungsi sebagai sistem komunikasi antara backend aplikasi dan pengguna dalam bentuk pemberitahuan real-time. Koleksi ini menyimpan data seperti

notificationId, userId, title, message, dan timestamp, yang semuanya digunakan untuk menampilkan pesan kontekstual kepada pengguna. Sistem ini berperan penting dalam menjaga interaksi yang aktif dan informatif selama proses pemesanan makanan berlangsung.

Setiap notifikasi dikirim secara otomatis oleh sistem berdasarkan peristiwa tertentu, seperti ketika pesanan telah diproses atau siap untuk diambil. Pengiriman pesan dilakukan melalui Firebase Cloud Messaging (FCM), dan entry pada koleksi ini diciptakan oleh Cloud Functions yang dipicu oleh perubahan status pesanan di koleksi orders. Hal ini memungkinkan proses notifikasi yang serverless, otomatis, dan terintegrasi langsung dengan sistem backend tanpa memerlukan intervensi manual dari admin atau tenant.

Selain notifikasi terkait pesanan, struktur ini juga dapat digunakan untuk menyampaikan pesan promosi dari tenant atau pengumuman penting dari admin kampus. Misalnya, tenant dapat mengirimkan diskon khusus, atau pihak kampus dapat menyampaikan notifikasi operasional seperti perubahan jam buka kantin. Dengan menyimpan semua notifikasi dalam koleksi tersendiri, sistem dapat menampilkan riwayat notifikasi di dalam aplikasi dan memberikan pengalaman pengguna yang konsisten serta terdokumentasi.

Struktur ini juga mendukung pengaturan preferensi pengguna, seperti mematikan notifikasi tertentu atau hanya menerima jenis notifikasi yang diinginkan. Dengan demikian, pengguna memiliki kendali penuh atas pengalaman notifikasi mereka, dan sistem dapat meminimalisir gangguan yang tidak relevan. Gambar 3.5 menampilkan bagaimana pendekatan modular pada koleksi notifications memungkinkan komunikasi yang efektif, tepat sasaran, dan mendukung transparansi proses dalam aplikasi U-Teen

3.5.2 Kendala yang Ditemukan

Selama proses pengembangan sistem aplikasi dan database, beberapa kendala teknis dan adaptasi ditemukan, baik dari sisi pemahaman teknologi maupun pada proses implementasi fitur.

1. Adaptasi terhadap Teknologi Baru

Tantangan muncul pada tahap awal pengembangan, yaitu adaptasi terhadap bahasa pemrograman Dart dan framework Flutter, serta platform backend Firebase. Ketiganya merupakan teknologi baru yang belum pernah digunakan sebelumnya, sehingga diperlukan waktu dan pembelajaran tambahan untuk memahami arsitektur, sintaksis, dan pola kerja masing-masing.

2. Kesulitan Integrasi Firebase dengan Kode Aplikasi

Hambatan ditemukan dalam proses integrasi antar sistem, terutama dalam menghubungkan komponen frontend (Flutter) dengan database Firebase. Masalah seperti ketidakcocokan struktur data, kesalahan autentikasi, serta error pada saat pengambilan data menjadi tantangan utama selama proses sinkronisasi antar lapisan aplikasi.

3. Munculnya Bug saat Pengembangan Fitur Tambahan

Seiring dengan bertambahnya fitur yang dikembangkan, sistem mengalami peningkatan kompleksitas. Beberapa bug mulai muncul terutama ketika fitur baru seperti sistem rating, histori transaksi, dan notifikasi real-time diimplementasikan. Bug ini menyebabkan gangguan fungsional pada beberapa komponen aplikasi.

4. Kesulitan dalam Menyelesaikan Bug Kompleks

Terdapat kendala dalam proses debugging, khususnya untuk fitur yang melibatkan kombinasi Firebase Cloud Functions, Firestore, dan frontend. Masalah seperti keterlambatan update status pesanan, error

dalam query data, hingga duplikasi transaksi sempat terjadi dan memerlukan waktu lebih lama untuk diidentifikasi dan diperbaiki.

3.5.3 Solusi atas Kendala yang Ditemukan

Dalam pelaksanaan proyek MBKM Wirausaha yang berfokus pada pengembangan sistem database U-Teen, berbagai kendala teknis dan adaptasi yang dihadapi memerlukan strategi penyelesaian yang spesifik, terencana, dan berkelanjutan. Proses penyelesaian kendala dilakukan secara progresif, melalui pendekatan pembelajaran mandiri, eksplorasi alat bantu debugging, serta kolaborasi internal dalam tim pengembang.

Berikut ini merupakan solusi-solusi utama yang diterapkan untuk mengatasi kendala yang ditemukan selama pelaksanaan pengembangan sistem:

1. Pendalaman Materi Melalui Dokumentasi Resmi dan Pembelajaran Mandiri

Kendala adaptasi terhadap teknologi baru, seperti bahasa pemrograman Dart, framework Flutter, dan platform Firebase, diatasi dengan melakukan pembelajaran mandiri secara intensif. Proses pembelajaran difokuskan pada:

1. Membaca dokumentasi resmi dari <https://firebase.google.com> dan <https://flutter.dev>, untuk memahami struktur arsitektur, cara implementasi, dan best practice dalam penggunaan layanan Firebase serta widget Flutter.
2. Mengikuti video tutorial terstruktur di platform pembelajaran seperti YouTube (misalnya Fireship, The Net Ninja, dan FlutterFlow), yang memaparkan skenario nyata dalam penggunaan Firebase Authentication, Firestore CRUD operations, hingga proses binding data menggunakan StreamBuilder di Flutter.
3. Melakukan latihan kecil secara terpisah (mini project) di luar sistem U-Teen, dengan tujuan untuk membiasakan diri dengan

fitur-fitur dasar Firebase seperti login user, input data, dan menampilkan data ke tampilan UI.

Melalui strategi ini, proses adaptasi terhadap teknologi baru menjadi lebih cepat dan efektif. Pemahaman terhadap ekosistem Firebase dan Flutter dapat dibangun secara bertahap hingga mencapai tingkat penguasaan yang cukup untuk diterapkan dalam proyek skala nyata.

2. Penggunaan Firebase Console dan Emulator sebagai Alat Debugging

Kesulitan dalam menghubungkan Firebase dengan sistem aplikasi, khususnya dalam sinkronisasi data antara Firestore dengan tampilan Flutter, diselesaikan melalui eksplorasi alat bantu yang tersedia di Firebase Console dan emulator aplikasi.

Firebase Console dimanfaatkan secara optimal sebagai:

1. Alat visualisasi struktur database, untuk memverifikasi apakah data berhasil tersimpan dan ditulis dengan format yang benar.
2. Sumber pengecekan real-time, untuk memastikan bahwa update status pesanan dan transaksi dikirim dan diterima dengan benar oleh koleksi yang dituju.
3. Pemantauan log dan error, terutama untuk mengidentifikasi kesalahan seperti kegagalan autentikasi, duplikasi dokumen, atau ketidakcocokan path data.

Selain itu, penggunaan emulator Flutter untuk Android Studio atau Visual Studio Code juga membantu tim dalam melihat langsung bagaimana UI merespons setiap perubahan pada data yang dikirim dari atau ke Firebase. Uji coba dilakukan dengan skenario seperti:

1. Membuat pesanan dan melihat apakah tenant menerima notifikasi secara real-time,

2. Memeriksa apakah status pesanan berubah setelah diklik oleh tenant,
3. Melihat tampilan kesalahan jika user belum login atau jika data kosong.

Dengan pendekatan ini, integrasi antara sistem Firebase dan komponen aplikasi dapat diperbaiki secara bertahap. Bug yang sebelumnya sulit ditelusuri dapat ditemukan lebih cepat dan diselesaikan dengan tepat.

3. Pengujian Iteratif dan Uji Coba Berkala

Masalah teknis yang muncul akibat penambahan fitur dan peningkatan kompleksitas aplikasi diatasi dengan melakukan proses pengujian berulang (iteratif testing). Setelah setiap fitur baru dikembangkan, uji coba langsung dilakukan dalam bentuk skenario nyata.

Contoh skenario pengujian yang dilakukan meliputi:

1. Pengguna melakukan pemesanan dari akun konsumen lalu tenant menerima pesanan kemudian tenant mengubah status pesanan dan konsumen menerima notifikasi.
2. Tenant menambahkan menu baru lalu menu tampil di dashboard konsumen Kemudian konsumen akan melakukan pemesanan dan transaksi tercatat di database.
3. Login dari berbagai akun dengan peran berbeda (admin, tenant, konsumen) lalu pengujian otorisasi akses data sesuai role.

Hasil dari pengujian ini didokumentasikan secara sistematis, lalu dibandingkan dengan hasil yang diharapkan. Jika terdapat ketidaksesuaian, perbaikan segera dilakukan. Pendekatan ini memungkinkan deteksi bug sejak dini, serta meminimalkan risiko kerusakan sistem akibat integrasi fitur baru.

4. Kolaborasi Tim dan Review Lintas Peran

Sebagian kendala kompleks, khususnya yang berkaitan dengan alur data lintas komponen, diselesaikan melalui diskusi teknis antaranggota tim. Kolaborasi ini dilakukan secara berkala melalui pertemuan mingguan dan sesi review kode. Melalui sesi kolaborasi, setiap anggota tim menyampaikan kendala yang dihadapi dan mendapatkan masukan dari sudut pandang berbeda, misalnya:

1. Perbaikan pada sisi UI oleh tim desain untuk mengurangi error input pengguna,
2. Perubahan alur navigasi atau pengaturan loading agar aplikasi lebih responsif,
3. Meninjau ulang skema database agar lebih efisien dan tidak menimbulkan query lambat.

Pendekatan kolaboratif ini juga membantu mempercepat pemahaman lintas peran, sehingga solusi yang diambil dapat menyelesaikan masalah dari akarnya, bukan hanya pada tampilan luar.

