

Pengembangan Aplikasi Android Untuk Management Coffee Shop

Alex Assalam^{1*}, Ivana Lucia Kharisma², Kamdan³

¹Program Studi Teknik Informatika, Fakultas Teknik, Komputer dan Desain

Universitas Nusa Putra Sukabumi

e-mail: alex.assalam_ti20@nusaputra.ac.id, ivana.lucia@nusaputra.ac.id,

kamdan@nusaputra.ac.id

Abstrak - Dalam menghadapi tantangan globalisasi dan perkembangan industri, aplikasi Android untuk manajemen coffee shop menjadi solusi penting guna meningkatkan efisiensi operasional dan memberikan pengalaman pelanggan yang lebih baik. Penelitian ini bertujuan untuk mengidentifikasi kebutuhan pengguna, mengembangkan aplikasi Android yang efisien, dan mengevaluasi manfaatnya dalam konteks manajemen coffee shop. Melalui studi literatur, analisis kebutuhan pengguna, dan penelitian persaingan, kami merancang aplikasi yang melibatkan fitur-fitur utama seperti manajemen stok, pemesanan, pembayaran, dan analisis data. Proses pengembangan melibatkan uji coba prototipe, pengujian fungsionalitas, dan integrasi dengan perangkat pendukung. Aplikasi yang dihasilkan bertujuan untuk meningkatkan efisiensi operasional dengan otomatisasi tugas-tugas kritis, meningkatkan ketepatan manajemen keuangan, dan memberikan pengalaman pelanggan yang lebih baik melalui proses pemesanan yang efisien. seperti memfasilitasi layanan pesan antar dan mengelola kapasitas pelanggan dengan lebih efektif. Dalam evaluasi oleh pengguna, aplikasi menunjukkan hasil positif dengan peningkatan efisiensi, ketepatan keuangan, dan kepuasan pelanggan. Manfaat lain meliputi analisis data yang mendalam untuk mendukung pengambilan keputusan strategis dan ketersediaan informasi real-time yang memungkinkan respon cepat terhadap perubahan situasi bisnis. Hasil penelitian ini memberikan kontribusi pada perkembangan aplikasi Android yang berfokus pada kebutuhan spesifik industri coffee shop. Kata Kunci - Komponen : Coffee Shop, Sejarah, Peran Sosial, Budaya

Abstract - In facing the challenges of globalization and industrial development, Android applications for coffee shop management are an important solution to increase operational efficiency and provide a better customer experience. This research aims to identify user needs, develop an efficient Android application, and evaluate its benefits in the context of coffee shop management. Through literature study, user needs analysis and competitive research, we designed an application involving key features such as stock management, ordering, payments and data analysis. The development process involves prototype testing, functionality testing, and integration with supporting tools. The resulting application aims to increase operational efficiency by automating critical tasks, increasing the accuracy of financial management, and providing a better customer experience through an efficient ordering process. such as facilitating delivery services and managing customer capacity more effectively. In evaluations by users, the application shows positive results with increased efficiency, financial accuracy and customer satisfaction. Other benefits include in-depth data analysis to support strategic decision making and the availability of real-time information that enables rapid response to changing business situations. The results of this research contribute to the development of Android applications that focus on the specific needs of the coffee shop industry.

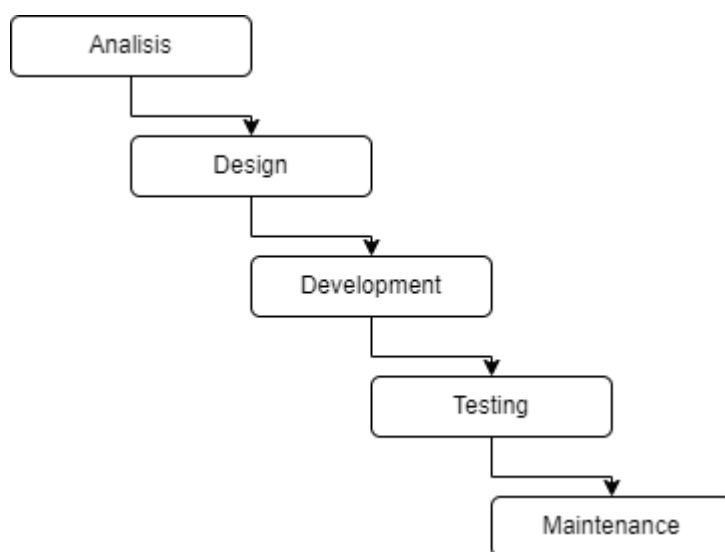
Keywords - Components: Coffee Shop, History, Social Role, Culture

I. PENDAHULUAN

Dalam era sekarang digitalisasi banyak di terapkan di berbagai sektor bisnis dari yang mendasar maupun cara beroperasi. Tidak terkecuali untuk bisnis coffee shop yang menuntut efisiensi dalam operasional, peningkatan pelayanan dan keberlangsungan bisnis coffee shop itu sendiri. Seiring berkembangnya bisnis, pengelola coffee shop sering kali menghadapi tugas-tugas kompleks, seperti manajemen inventaris bahan mentah, manajemen inventaris, dan pelacakan penjualan. Pengembangan aplikasi dapat membantu meningkatkan efisiensi operasional dengan mengotomatiskan proses-proses ini. Pelanggan mengharapkan pengalaman yang lebih baik dan layanan yang cepat. Aplikasi dapat mempermudah pemesanan, pembayaran, dan pemberian masukan. Hal ini dapat meningkatkan kepuasan pelanggan dan membangun loyalitas. Pengelolaan keuangan yang tepat dan akurat merupakan kunci keberhasilan dalam berbisnis. Aplikasi ini dapat membantu mencatat penjualan, pembelian, dan pelaporan keuangan secara real-time sehingga memudahkan pengelolaan keuangan. Aplikasi dapat memberikan analisis data yang mendalam mengenai kinerja penjualan, preferensi pelanggan, dan tren pasar. Hal ini akan membantu manajemen mengambil keputusan yang lebih baik untuk perkembangan bisnis di masa depan.

II. METODE

Metode penelitian dalam pengembangan aplikasi menggunakan metode waterfall. Metode waterfall itu sendiri adalah salah satu model pengembangan perangkat lunak (SDLC) yang paling awal dan paling umum digunakan. Model ini menggunakan pendekatan sistematis dan berurutan, di mana setiap tahapan harus diselesaikan sebelum melanjutkan ke tahapan selanjutnya. Model Waterfall sering digunakan untuk proyek pengembangan perangkat lunak yang relatif kecil dan sederhana, karena setiap tahapannya mengalir seperti air terjun dan hanya dapat dimulai setelah tahapan sebelumnya selesai. Model ini memiliki persyaratan yang jelas dan tidak berubah.



Gambar 1 Metode Waterfall

a. Analisis Kebutuhan

Analisis adalah fase awal siklus pengembangan di mana persyaratan sistem atau perangkat lunak ditentukan dan dianalisis secara cermat. Analisis ini bertujuan untuk memahami sepenuhnya kebutuhan pengguna dan bisnis, sehingga dapat dibuat spesifikasi yang sesuai sebelum memasuki tahap desain dan implementasi.

- **Pengumpulan Kebutuhan**
Tim proyek berinteraksi dengan pemangku kepentingan, termasuk pengguna akhir, manajemen, dan pihak berkepentingan lainnya. Informasi dan kebutuhan masing-masing pihak dikumpulkan secara rinci untuk memahami pandangan dan harapan mereka terhadap sistem yang akan dikembangkan.
- **Penentuan Kebutuhan Fungsional dan Non Fungsional**
Persyaratan fungsional berhubungan dengan fitur dan fungsi yang diharapkan dari perangkat lunak. Persyaratan non-fungsional termasuk aspek seperti keamanan, kinerja, dan skalabilitas harus dipertimbangkan.
- **Dokumentasi Kebutuhan**
Semua kebutuhan yang diidentifikasi secara rinci didokumentasikan sebagai dokumen spesifikasi kebutuhan perangkat lunak. Dokumen ini mencakup penjelasan rinci tentang persyaratan fungsional dan non-fungsional serta skenario penggunaan dan batasan sistem.
- **Validasi Kebutuhan**
Dokumen persyaratan divalidasi bersama pihak terkait untuk memastikan seluruh persyaratan penting dan relevan terpenuhi. Setiap ketidakjelasan atau ketidaksesuaian diidentifikasi dan diselesaikan sebelum melanjutkan ke langkah berikutnya.
- **Penyusunan Dalam Bentuk Rencana Proyek**
Hasil analisis kebutuhan menjadi dasar untuk mengembangkan rencana proyek yang mencakup alokasi sumber daya, perkiraan biaya, dan jadwal yang realistis.
- **Pengambilan Keputusan**
Analisis persyaratan membantu membuat keputusan strategis mengenai arsitektur sistem, pemilihan teknologi, dan strategi pengembangan yang akan digunakan.

b. Perancangan

Pada tahap ini, tim pengembangan perangkat lunak merinci kebutuhan yang dikumpulkan pada tahap pertama menjadi desain teknis yang lebih rinci. Beberapa aspek yang dibahas pada tahap ini antara lain.

- **Arsitektur Sistem**
Tim merancang keseluruhan struktur sistem perangkat lunak. Hal ini mencakup bagaimana komponen perangkat lunak akan berinteraksi satu sama lain dan bagaimana data akan mengalir di antara komponen-komponen tersebut. Penentuan arsitektur sistem menjadi dasar implementasi selanjutnya.
- **Desain Database**
Jika aplikasi melibatkan penggunaan database, maka pada langkah ini tim akan merinci skema database yang akan digunakan. Ini mencakup tabel, hubungan antar tabel, dan skema lain yang diperlukan untuk menyimpan dan mengelola data.
- **Desain Antarmuka Pengguna (UI/UX)**
Tim merancang antarmuka pengguna yang akan digunakan oleh pengguna akhir. Ini melibatkan pemikiran tentang tata letak elemen antarmuka, navigasi, dan aspek lain yang memengaruhi pengalaman pengguna.

- **Desain Program (Coding)**
Pada tingkat yang lebih terperinci, tim merinci cara mengimplementasikan setiap komponen perangkat lunak. Hal ini mungkin termasuk memilih algoritma, struktur data, dan logika kontrol yang diperlukan untuk mengimplementasikan fungsionalitas yang diinginkan.
- **Desain Keamanan**
Aspek yang terkait dengan keamanan sistem juga diperhitungkan sejak tahap desain. Tim merencanakan langkah-langkah keamanan yang diperlukan untuk melindungi sistem dari potensi ancaman.
- **Desain Pengujian**
Desain pengujian (testing) juga disiapkan pada tahap ini. Hal ini mencakup pengujian unit, pengujian integrasi, dan pengujian sistem untuk memastikan bahwa perangkat lunak beroperasi sesuai dengan kebutuhan yang telah ditentukan.

c. Implementasi

Fase di mana tim pengembang mulai menulis kode sumber untuk membuat produk perangkat lunak yang sesuai dengan spesifikasi desain. Ini merupakan tahap ketiga metode Waterfall. Ini adalah prosedur umum yang terlibat dalam tahap implementasi:

- **Pemilihan Teknologi dan Alat**
Tim pengembangan memilih alat pengembangan, framework, dan bahasa pemrograman yang sesuai dengan spesifikasi desain dan kebutuhan proyek. Pemilihan teknologi yang akan digunakan untuk mengimplementasikan komponen tertentu dari aplikasi termasuk dalam keputusan ini.
- **Penulisan Kode Sumber**
Setelah desain disetujui, pengembang mulai menulis kode sumber. Logika bisnis, manipulasi data, antarmuka pengguna, dan komponen lainnya yang diperlukan disertakan dalam kode ini.
- **Pengujian unit**
Setiap modul atau bagian aplikasi diuji secara terpisah untuk memastikan bahwa semua komponen bekerja dengan benar. Tujuan pengujian unit adalah untuk memastikan bahwa setiap komponen kode memenuhi persyaratan fungsional yang telah ditentukan.
- **Integrasi Kode**
Setelah pengujian unit berhasil, modul digabungkan untuk melakukan pengujian integrasi untuk memastikan bahwa berbagai komponen aplikasi dapat berinteraksi dan berkomunikasi dengan benar.
- **Pemecahan Masalah (Debugging)**
Pengembang melakukan debugging untuk menemukan dan memperbaiki kesalahan atau bug selama tahap pengujian unit atau integrasi.
- **Dokumentasi Kode**
Pengembang juga membuat dokumentasi kode selama atau setelah penulisan kode. Dokumentasi ini mencakup komentar di dalam kode, penjelasan tentang fungsi, dan membantu pengembang lain atau pemeliharaan di masa depan. Selama atau setelah penulisan kode, pengembang juga membuat dokumentasi kode. Dokumentasi ini mencakup komentar-komentar di dalam kode, menjelaskan fungsi-fungsi, dan memberikan panduan bagi pengembang lain atau untuk pemeliharaan di masa depan.

- **Optimasi Kode**
Optimasi kode dapat dilakukan oleh tim untuk meningkatkan kinerja aplikasi, mengurangi konsumsi sumber daya, atau mempercepat waktu eksekusi. Ini dapat mencakup peningkatan efisiensi, struktur kode, atau pengoptimalan algoritma.
- **Pengujian Fungsional**
Aplikasi diuji secara menyeluruh setelah integrasi selesai untuk memastikan bahwa fungsinya memenuhi persyaratan dan spesifikasi.

d. Testing

Selama tahap kebutuhan dan desain, sistem atau aplikasi yang diimplementasikan diuji secara menyeluruh untuk memastikan bahwa ia memenuhi persyaratan dan spesifikasi yang ditetapkan.

- **Pengujian Unit (Unit Testing)**
Ini adalah fase di mana setiap modul atau unit perangkat lunak diuji secara individual. Unit, komponen terkecil dari perangkat lunak yang dapat diuji secara mandiri, seperti metode atau fungsi, dimaksudkan untuk memastikan bahwa setiap unit atau modul perangkat lunak beroperasi sesuai dengan desain.
- **Pengujian Integrasi (Integration Testing)**
Setelah pengujian unit, modul-modul digabungkan untuk membentuk kumpulan yang lebih besar. Untuk memastikan bahwa mereka dapat berfungsi secara sinergis, pengujian integrasi menguji bagaimana modul-modul berinteraksi satu sama lain.
- **Pengujian Sistem (System Testing)**
Pada tahap ini, sistem secara keseluruhan diuji untuk memastikan bahwa aplikasi beroperasi sesuai dengan spesifikasi. Pengujian ini mencakup berbagai skenario penggunaan dan kondisi kerja yang mungkin terjadi di lingkungan produksi.
- **Pengujian Penerimaan Pengguna (User Acceptance Testing - UAT)**
Setelah tahap pengujian sistem selesai, aplikasi dapat dikirim ke pengguna akhir atau pemangku kepentingan untuk diuji penerimaan pengguna. UAT dilakukan untuk memastikan bahwa aplikasi memenuhi kebutuhan bisnis dan memenuhi ekspektasi pengguna.
- **Pemecahan Masalah (Debugging) dan Perbaikan Bug**
Tim pengembangan melakukan debugging untuk mengidentifikasi dan memperbaiki kesalahan atau bug selama tahap pengujian. Bug ini harus diperbaiki sebelum melanjutkan ke tahap berikutnya.
- **Verifikasi dan Validasi**
Proses verifikasi dan validasi memastikan bahwa perangkat lunak dibangun sesuai dengan spesifikasi dan memenuhi kebutuhan pengguna dan kebutuhan bisnis.

e. Deployment

Dalam metode Waterfall, tahap Deployment (Implementasi) adalah tahap di mana sistem atau aplikasi yang telah dikembangkan dan diuji secara menyeluruh diterapkan ke lingkungan produksi. Ini mencakup proses memasukkan atau merilis aplikasi ke penggunaan aktif, yang memungkinkan pengguna akhir untuk mengakses dan menggunakannya.

- **Pemilihan Lingkungan Produksi**
Untuk menjalankan aplikasi dengan efisien, tim pengembangan dan operasional memilih dan menyiapkan lingkungan produksi yang tepat. Ini termasuk memilih server, database, dan infrastruktur lainnya yang sesuai dengan kebutuhan aplikasi.

- **Konfigurasi Server dan Infrastruktur**
Tim sistem mengkonfigurasi server dan infrastruktur sesuai dengan persyaratan teknis aplikasi setelah memilih lingkungan produksi. Ini termasuk instalasi dan konfigurasi perangkat lunak, pengaturan parameter sistem, dan penyiapan komponen lainnya.
- **Migrasi Data**
Tim pengembangan atau tim basis data akan dengan hati-hati memindahkan data dari lingkungan pengembangan atau pengujian untuk memastikan integritas dan konsistensi data.
- **Pengujian Akhir**
Pengujian akhir dilakukan di lingkungan produksi untuk memastikan bahwa aplikasi berjalan dengan baik dan memenuhi spesifikasi. Ini termasuk pengujian fungsional dan kinerja untuk memastikan bahwa semua fitur beroperasi dengan benar.
- **Pelatihan Pengguna**
Untuk memastikan pengguna akhir memahami cara menggunakan aplikasi dengan benar, pelatihan pengguna dapat dilakukan jika diperlukan. Ini dapat mencakup pelatihan langsung, panduan pengguna, atau sumber daya pelatihan lainnya.
- **Pemulihan Darurat**
Data dan konfigurasi sistem diperbarui, dan prosedur pemulihan darurat dan backup diatur untuk mengantisipasi dan mengatasi masalah atau kegagalan yang dapat terjadi selama atau setelah implementasi.
- **Peluncuran Aplikasi**
Aplikasi secara resmi diunggah atau diinstal ke lingkungan produksi, yang mencakup penyebaran file aplikasi ke server, konfigurasi akhir, dan semua langkah teknis yang diperlukan untuk menjalankan aplikasi di lingkungan produksi.
- **Pemantauan dan Dukungan Awal**
Setelah implementasi, tim dukungan dan operasional secara aktif memantau kinerja aplikasi. Segera setelah peluncuran, dukungan awal tersedia untuk membantu pengguna akhir dan memecahkan masalah mereka.
- **Dokumentasi Pemeliharaan**
Dokumentasi pemeliharaan diperbarui dengan informasi terbaru, yang mencakup konfigurasi lingkungan produksi dan tindakan pemeliharaan yang mungkin diperlukan. Informasi ini sangat penting untuk pedoman di masa depan.

f. **Pemeliharaan**

Tahap Pemeliharaan (Maintenance). Tahap ini terjadi setelah sistem telah diimplementasikan dan dijalankan secara penuh.

- **Pemeliharaan Perbaikan**
Jika kesalahan atau bug sistem ditemukan setelah implementasi, tugas pemeliharaan ini melibatkan memperbaiki kesalahan tersebut. Jika ada masalah dengan penggunaan sistem, tim pengembang akan menanggapi dan memperbaikinya.
- **Pemeliharaan Adaptif**
Jika terjadi perubahan pada lingkungan atau perangkat keras, sistem mungkin perlu disesuaikan untuk menyesuaikan diri. Ini adalah bagian dari tanggung jawab pemeliharaan adaptif.

- **Pemeliharaan Perfective**
Dalam beberapa situasi, pelanggan atau pengguna akhir dapat meminta peningkatan kinerja atau fungsionalitas sistem. Pemeliharaan perfective adalah proses meningkatkan sistem tanpa memperbaiki kesalahan.
- **Pemeliharaan Preventif**
Pemeliharaan preventif dilakukan untuk mencegah kesalahan atau masalah di masa depan. Ini dapat mencakup membuat perubahan kecil atau meningkatkan untuk mencegah masalah.
- **Evaluasi Kinerja**
Setelah sistem digunakan, tim pengembang dapat mengevaluasi kinerjanya. Tujuannya adalah untuk memastikan sistem beroperasi dengan baik dan memenuhi kebutuhan pengguna.
- **Dokumentasi Pemeliharaan**
Memperbarui dokumentasi sistem selama tahap pemeliharaan sangat penting. Ini harus mencakup informasi tentang perubahan yang telah dilakukan, catatan tentang pemecahan bug, dan informasi lainnya yang mungkin diperlukan oleh pengembang atau tim dukungan di masa depan.

DAFTAR PUSTAKA

- [1]Kadek, N., Dewi, C., Bagus, I., Anandita, G., Atmaja, K. J., Aditama, P. W., & Magister, P. S. (2018). RANCANG BANGUN APLIKASI MOBILE SISK BERBASIS ANDROID. In SINTECH Journal | 100 SINTECH JOURNAL (Vol. 1, Issue 2). Online. <http://jurnal.stiki-indonesia.ac.id/index.php/sintechjournal>
- [2]Ependi, U., Panjaitan, F., Syakti, F., & Korespondensi, P. (2020). PENGEMBANGAN APLIKASI MOBILE TRAVEL GUIDE PADA PROVINSI SUMATERA SELATAN. 7(3), 607–618. <https://doi.org/10.25126/jtiik.202072107>
- [3]Surya Wardhana, W., Tolle, H., & Kharisma, A. P. (2019). Pengembangan Aplikasi Mobile Transaksi Bank Sampah Online Berbasis Android (Studi Kasus: Bank Sampah Malang) (Vol. 3, Issue 7). <http://j-ptiik.ub.ac.id>
- [4]Hendryani, A., Teknik, J., Politeknik, E., & Kemenkes, K. (2020). PENGEMBANGAN APLIKASI MOBILE HEALTH BERBASIS ANDROID UNTUK MONITORING DAN EVALUASI STUNTING. Jurnal Sehat Mandiri, 15. <http://jurnal.poltekkespadang.ac.id/ojs/index.php/jsm>
- [5]Brata, K. C., & Brata, A. H. (2018). Pengembangan Aplikasi Mobile Augmented Reality untuk Mendukung Pengenalan Koleksi Museum. Jurnal Teknologi Informasi Dan Ilmu Komputer, 5(3), 347–352. <https://doi.org/10.25126/jtiik.201853798>
- [6]Nopriansyah, U. (2020). PENGEMBANGAN APLIKASI KESEHATAN BERBASIS MOBILE UNTUK PEMANTAUAN DETEKSI DINI TUMBUH KEMBANG (DDTK) ANAK USIA 4-6 TAHUN. 3(1), 98–111. <http://ejournal.radenintan.ac.id/index.php/al-athfaal>
- [7]Rolly, N., & Hakiem, N. (2015). BERBASIS ANDROID UNTUK PENGGUNA DOSEN DAN MAHASISWA (Studi Kasus : Pusat Teknologi Informasi dan Pangkalan Data (Pustipanda) UIN Syarif Hidayatullah Jakarta) (Vol. 8, Issue 1). AIS.