

Pengembangan Aplikasi Monitoring Anjungan Tunai Mandiri (ATM) di PT. Collega Inti Pratama

1st Arham Fuzi
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

arhamfauzi@student.telkomuniversity.a
c.id

2nd Fery Prasetyanto
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

ferypras@telkomuniversity.ac.id

3rd Desy Puspa Rahayu
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

pusparahayu@telkomuniversity.ac.id

Abstrak — Penelitian ini menanggapi meningkatnya kebutuhan masyarakat akan layanan perbankan melalui Automated Teller Machine (ATM) dan Cash Recycling Machine (CRM) dengan mengembangkan aplikasi Monitoring ATM Mobile. Tujuan utama penelitian adalah memastikan ketersediaan, kinerja optimal, dan responsivitas layanan perangkat tersebut. Metode pengembangan yang digunakan adalah metode waterfall, yang mencakup analisis kebutuhan, desain, implementasi, pengujian, dan deployment. Aplikasi Monitoring ATM Mobile dapat memonitor kondisi perangkat ATM dan CRM secara cepat, mengoptimalkan ketersediaan layanan, dan meningkatkan responsivitas terhadap masalah. Sistem pemantauan dan pelaporan maintenance memungkinkan teknisi untuk melaporkan dan menangani masalah secara langsung melalui smartphone Android, meningkatkan efisiensi operasional. Aplikasi dashboard berbasis client-server menyediakan backend yang dapat untuk mengelola dan memeriksa daftar permasalahan kerusakan, memudahkan admin dalam pengelolaan dan pemantauan kondisi perangkat secara keseluruhan. Dengan demikian, penelitian ini berhasil mencapai tujuannya untuk menyediakan solusi yang cepat dan mudah bagi pemantauan dan pelaporan kondisi perangkat ATM dan CRM. Implementasi teknologi Flutter dalam pengembangan aplikasi juga mendukung responsivitas aplikasi mobile ini dalam memenuhi kebutuhan masyarakat akan layanan perbankan modern.

Kata Kunci: Automated Teller Machine (ATM), Mobile Application, Client-server, Flutter.

I. PENDAHULUAN

Seiring dengan meningkatnya kebutuhan masyarakat terhadap layanan perbankan melalui terminal Automated Teller Machine (ATM) dan Cash Recycling Machine (CRM), pemerintah dan lembaga keuangan berupaya menyediakan akses mudah dengan memasang terminal di lokasi strategis. Bertambahnya perangkat ini memunculkan kebutuhan untuk memastikan kinerja optimal dan ketersediaan perangkat. Oleh karena itu, pemantauan cepat dan mudah menjadi penting. PT Collega Inti Pratama, perusahaan TI nasional dalam Telkom Group, menyadari perlunya aplikasi ATM Monitoring berbasis mobile (Android dan iOS) dan dashboard berbasis client-server sebagai backend untuk memeriksa masalah yang teridentifikasi. Sistem ini memungkinkan kegiatan maintenance dilaporkan secara online melalui smartphone Android, memudahkan pemantauan dan pelaporan secara efisien, mendukung pelayanan yang cepat dan responsif. Tujuan dari proyek ini

adalah menciptakan aplikasi monitoring yang cepat dan mudah digunakan untuk memantau kondisi ATM dan CRM, serta mengembangkan sistem pelaporan maintenance online yang terintegrasi dengan dashboard client-server untuk pemeriksaan masalah.

II. KAJIAN TEORI

A. Mobile Application

Mobile Application atau Aplikasi Mobile merupakan sebuah aplikasi yang dibuat khusus untuk platform mobile seperti iOS, android atau windows mobile. Aplikasi mobile, atau biasa disebut mobile apps, merupakan aplikasi internet yang dirancang khusus untuk dijalankan pada smartphone, tablet, dan perangkat mobile lainnya. [1].

B. Automatic Teller Machine (ATM) Monitoring

Monitoring ATM adalah aplikasi yang dirancang untuk memantau kondisi dan kinerja mesin ATM. Aplikasi ini memungkinkan pengguna untuk melihat informasi real-time tentang status terminal ATM, ketersediaan uang, masalah perangkat keras, dan koneksi jaringan, serta memberikan sistem tiket untuk pemeliharaan preventif dan korektif. [2].

C. Client-Server Architecture

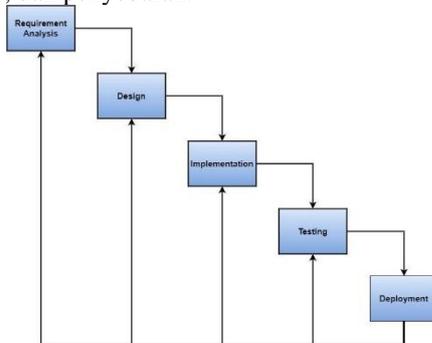
Client-Server Architecture adalah model komputasi jaringan di mana tugas dibagi antara dua jenis komputer. Yang pertama Client, komputer ini mengajukan permintaan untuk sumber daya atau data. Sedangkan Server merupakan komputer yang menyediakan sumber daya atau data ke Client sebagai tanggapan atas permintaannya. [3].

D. Flutter

Flutter adalah kerangka pengembangan aplikasi mobile yang diperkenalkan pada “Google Developer Days” tahun 2018. Flutter merupakan framework pengembangan aplikasi mobile open source dan gratis yang dikembangkan dan didukung oleh Google. Flutter digunakan untuk membangun aplikasi Android dan iOS, serta menjadi metode utama untuk membuat aplikasi untuk sistem operasi Google Fuchsia (Developers,2018; Singh & Bhardwaj,2019). [4].

III. METODE

Penulis menggunakan metode waterfall dalam pengembangan aplikasi Monitoring ATM Mobile karena metode ini merupakan pendekatan umum dalam pengembangan perangkat lunak yang menawarkan proses yang terstruktur dan terorganisir dengan baik. Keunggulan metode ini termasuk kemudahan dalam memahami struktur dan dokumentasi yang baik. Metode waterfall terdiri dari lima tahapan: analisis kebutuhan, desain, implementasi, pengujian, dan penyebaran.



A. Requirement Analysis

Penulis melakukan analisis kebutuhan untuk mengidentifikasi apa saja yang diperlukan dalam implementasi aplikasi Monitoring ATM Mobile. Analisis ini mencakup kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional meliputi fitur-fitur yang harus ada dalam aplikasi atau user requirement. Sementara itu, kebutuhan non-fungsional mencakup aspek-aspek yang mendukung tampilan, fungsi, dan properti aplikasi.

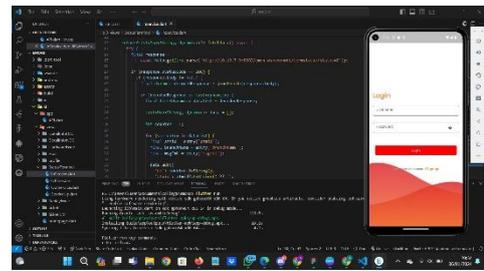
B. Design

Setelah memahami kebutuhan, penulis yang menggunakan Metode *Waterfall* merancang desain sketsa tampilan antarmuka pengguna dan diagram aplikasi. Penulis desain tampilan *user interface* ini menggunakan figma.



C. Implementation

Tahapan berikutnya adalah implementasi, di mana fokusnya adalah pembuatan kode program dan pengujian untuk memastikan kualitas perangkat lunak yang dibangun yaitu aplikasi Monitoring ATM Mobile.



D. Testing

Pengujian aplikasi Monitoring ATM Mobile dilakukan menggunakan dua metode: blackbox testing dan whitebox testing. Blackbox testing memastikan semua fitur dan fungsionalitas berjalan sesuai spesifikasi dengan melibatkan berbagai pengguna untuk mendapatkan umpan balik objektif. Whitebox testing, di sisi lain, menganalisis dan memeriksa struktur internal serta kode aplikasi untuk menilai aliran dan proses di dalam perangkat lunak, memastikan bahwa kode dan struktur internal mendukung fungsionalitas yang diinginkan.

E. Deployment

Deployment mencakup proses penyebaran aplikasi ke lingkungan produksi. Setelah pengujian selesai, aplikasi dipersiapkan untuk peluncuran, termasuk konfigurasi sistem, instalasi di server, dan integrasi dengan sistem yang ada. Tahap ini memastikan bahwa aplikasi siap digunakan secara luas oleh pengguna akhir dengan performa optimal.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Black Box Testing

Pengujian blackbox dilakukan untuk mengevaluasi apakah semua fitur dan fungsionalitas aplikasi Monitoring ATM Mobile sesuai dengan spesifikasi yang telah ditentukan. Selama pengujian, berbagai fitur seperti pelaporan kerusakan, pemantauan status terminal, dan notifikasi diperiksa. Hasilnya menunjukkan bahwa seluruh fitur berfungsi dengan baik, dan aplikasi memenuhi semua kebutuhan fungsional yang telah diidentifikasi. Pengujian melibatkan beberapa pengguna dengan berbagai latar belakang, dan umpan balik mereka mengonfirmasi bahwa aplikasi mudah digunakan dan berfungsi sesuai harapan tanpa adanya cacat yang signifikan. Seperti: Tabel 1(A)

Tabel 1
(A)

BLACKBOX TESTING HALAMAN LOGIN					
No.	Skenario Pengujian	Test care	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Username, password tidak diisi tetapi klik button login	Username dan password (kosong)	Hasil login sistem gagal dan memberikan alert "Username or password is incorrect"	Hasil login sistem gagal dan memberikan alert "Username or password is incorrect"	Valid
2.	Mengetikkan username, tetapi tidak memasukkan password	Username (user) Password (kosong)	Hasil login sistem gagal dan memberikan alert "Username or password is incorrect"	Hasil login sistem gagal dan memberikan alert "Username or password is incorrect"	Valid
3.	Mengetikkan username, password tetapi tidak sesuai	Username (abc) Password (321)	Hasil login sistem gagal dan memberikan alert "Username or password is incorrect"	Hasil login sistem gagal dan memberikan alert "Username or password is incorrect"	Valid
4.	Mengetikkan username, password dengan benar	Username (user) Password (123)	Hasil login sistem berhasil dan masuk kedalam halaman dashboard	Hasil login sistem berhasil dan masuk kedalam halaman dashboard	Valid

TABEL 2
(A)

Fungsi yang Diuji	Deskripsi	Input	Output yang Diharapkan
<pre>Future<List<Map<String, dynamic>>> fetchData() async { try { final response = await http.get(Uri.parse('https://emons-fa07b-default-rtdb.asia-southeast1.firebaseio.com/app/Dashboard/-00HZJKZv3Lq0_gxg01/1.json')); if (response.statusCode == 200) { if (response.body != null) { final dynamic decodedResponse = jsonDecode(response.body); print("Decoded Response: \$decodedResponse"); } } } catch (e) {} }</pre>	Memeriksa status terminal (In service, Service up, out service, Offline).	"InService", "ServiceUp", "OutService", "Offline"	Nilai boolean atau string yang sesuai dengan status terminal.
<pre>try { final response = await http.get(Uri.parse('https://emons-fa07b-default-rtdb.asia-southeast1.firebaseio.com/app/Dashboard/-00HZJKZv3Lq0_gxg01/12.json')); if (response.statusCode == 200) { if (response.body != null) { final dynamic decodedResponse = jsonDecode(response.body); print("Decoded Response: \$decodedResponse"); } } }</pre>	Memeriksa masalah perangkat keras (Fatal, Warning). Dan perangkat lunak (empty, low, full, good).	"empty", "Low", "Full", "Good", "Fatal", "Warning"	Nilai boolean atau string yang sesuai dengan status perangkat keras dan perangkat lunak.
<pre>Future<List<Map<String, dynamic>>> fetchData(String status) async { try { final response = await http.get(Uri.parse('https://emons-fa07b-default-rtdb.asia-southeast1.firebaseio.com/app/Dashboard/-00HZJKZv3Lq0_gxg01/5.json')); if (response.statusCode == 200) { if (response.body != null) { final dynamic decodedResponse = jsonDecode(response.body); } } } catch (e) {} }</pre>	Memeriksa status konektivitas host (Sign ON, Sign Off, Down).	"SignON", "SignOff", "Down"	Nilai boolean atau string yang sesuai dengan status konektivitas.

B. Hasil Pengujian White Box Testing

Pengujian whitebox dilakukan untuk menilai struktur internal dan kode aplikasi Monitoring ATM Mobile. Analisis meliputi pemeriksaan aliran data, kontrol logika, dan struktur kode. Hasilnya menunjukkan bahwa kode aplikasi telah diimplementasikan dengan baik dan sesuai dengan desain sistem. Struktur internal mendukung fungsionalitas yang diinginkan, dan tidak ditemukan kesalahan logika atau masalah integrasi yang signifikan. Pengujian ini memastikan bahwa aliran data dan proses internal aplikasi berjalan lancar dan tidak mempengaruhi performa atau keamanan aplikasi. Contoh: Tabel 2(A)

V. KESIMPULAN

Berdasarkan analisis dan implementasi yang dilakukan, proyek akhir ini berhasil mencapai tujuan yang ditetapkan. Aplikasi Monitoring ATM Mobile yang dikembangkan mampu memantau perangkat ATM dan CRM dengan cepat dan mudah, memastikan ketersediaan dan kinerja optimal serta mempercepat layanan kepada masyarakat. Sistem ini juga memudahkan pemantauan dan pelaporan maintenance secara online melalui smartphone berbasis Android, meningkatkan efisiensi operasional. Selain itu, aplikasi dashboard berbasis client-server menyediakan backend efektif untuk memantau dan mengelola data kerusakan, mempermudah administrasi dan teknisi. Secara keseluruhan, proyek ini berhasil memberikan solusi efisien untuk pemantauan dan pelaporan kondisi perangkat, memastikan layanan yang optimal.

REFERENSI

- [1] Emily Borom, "Study Offers Early Look at How Internet is Changing Daily Life," 2000.
- [2] Jane Lubis, Internet *User Behaviour*.: McMillan Publishing, 2001.
- [3] Spearman Roberts, Information System: Now and Tomorrow. Chicago: Adventure Press, 2009.
- [4] John Rokoko, Pseudo-2D Hidden Markov Model. New York: McGraw Hill, 2005.