

PENGEMBANGAN APLIKASI KASIR MOBILE YANG EFISIEN (STUDI KASUS: IMPLEMENTASI API GEMINI PADA FLUTTER)

Adinda Cahya Kamilla, Natasya Priyani, Viktor Handrianus Pranatawijaya, Nova Noor Kamala Sari

Teknik Informatika, Universitas Palangka Raya

Jalan Yos Sudarso, Palangka, Kec. Jekan Raya, Kota Palangka Raya, Kalimantan Tengah 74874

adindack@mhs.eng.upr.ac.id

ABSTRAK

Penelitian ini bertujuan mengembangkan sistem kasir berbasis *mobile* untuk *coffee shop* dengan fokus pada efektivitas waktu pengguna. Dengan metode *Waterfall* yang disederhanakan, pengembangan dilakukan secara bertahap, meminimalisir kesalahan, dan memastikan setiap fase berjalan sesuai rencana. Analisis penting untuk memahami kebutuhan sistem maupun pengguna dan mengidentifikasi komponen yang diperlukan, sementara desain bertujuan untuk merancang arsitektur sistem yang memenuhi kebutuhan yang telah diidentifikasi. Pengkodean dilakukan dengan menggunakan berbagai teknologi dan alat seperti Flutter, bahasa Dart, *query database*, kecerdasan buatan (AI), dan komponen pendukung lainnya. Pengujian dilakukan untuk memverifikasi dan memvalidasi aplikasi, khususnya fungsi *generate* deskripsi. Hasilnya menunjukkan bahwa API Gemini berfungsi baik dan menghasilkan deskripsi sesuai harapan. Implementasi ini memungkinkan deskripsi produk dihasilkan secara otomatis, mengurangi kebutuhan input manual, dan meningkatkan efisiensi operasional. Pengujian juga menunjukkan bahwa fungsi *generate* deskripsi bekerja cepat, menghasilkan deskripsi dalam waktu 2-5 detik tanpa masalah signifikan. Penggunaan teknologi ini membuktikan bahwa integrasi AI dalam aplikasi kasir *mobile* dapat meningkatkan kualitas layanan dan pengalaman pengguna di *coffee shop*. Optimasi berkelanjutan dan penambahan fitur baru disarankan untuk memastikan aplikasi tetap relevan dan berkinerja optimal.

Kata kunci : API, gemini AI, flutter, aplikasi mobile, kasir, coffee shop

1. PENDAHULUAN

Teknologi memainkan peran penting dalam perkembangan perusahaan saat ini, sehingga menjadi salah satu faktor penentu keberhasilan dalam memajukan bisnis[1].

Salah satu sektor yang mengalami dampak signifikan dari revolusi teknologi adalah industri restoran, terutama *coffee shop* atau warung kopi, yang seringkali berfungsi sebagai pusat kegiatan sosial dan profesional. Lebih dari sekadar tempat untuk menikmati kopi dan makanan ringan, *coffee shop* telah menjadi arena untuk penerapan inovasi teknologi yang meningkatkan pengalaman pelanggan dengan lebih efisien dan memuaskan.

Salah satu contohnya adalah teknologi berbasis web. Teknologi Informasi berbasis web diyakini dapat memudahkan pengelolaan organisasi, membuat pekerjaan lebih efektif dan efisien secara otomatis, serta mempercepat proses pengolahan data. Misalnya, di sektor bisnis seperti kedai kopi, atau *coffee shop* penggunaan teknologi berbasis web dapat membantu dalam manajemen inventaris, pemesanan online, dan analisis data pelanggan, sehingga meningkatkan operasional dan pelayanan..

Dalam konteks ini, sistem kasir berbentuk *mobile* menjadi solusi yang inovatif dan sesuai dengan tren zaman. Sistem kasir *mobile* adalah perangkat lunak atau aplikasi yang digunakan untuk mengelola proses transaksi penjualan di bisnis, termasuk di *coffee shop*. Sistem kasir *mobile* memungkinkan pengguna untuk melakukan transaksi dengan mudah dan cepat melalui

perangkat seluler mereka, seperti *smartphone* atau tablet.

Tujuan utama dari aplikasi kasir *mobile* untuk kedai kopi adalah untuk meningkatkan efisiensi dan kenyamanan dalam pengelolaan transaksi dan operasional harian. Aplikasi ini dirancang untuk mempermudah pencatatan penjualan, manajemen inventaris, pelacakan stok secara real-time dan pengelolaan keuangan. Selain itu, aplikasi kasir *mobile* memungkinkan pemilik dan karyawan kedai kopi untuk melayani pelanggan dengan lebih cepat dan akurat, mengurangi kesalahan dalam pencatatan transaksi, serta menyediakan laporan penjualan yang komprehensif untuk analisis bisnis yang lebih baik.

Dengan penambahan API AI seperti API Gemini, sistem kasir *mobile* dapat ditingkatkan dengan kecerdasan buatan. API, atau *Application Programming Interface*, adalah kumpulan instruksi program yang memfasilitasi pembangunan aplikasi perangkat lunak[2]. Fungsi API sendiri adalah untuk pertukaran data antar aplikasi/platform, dengan fungsi atau instruksi tersendiri sehingga pengembang aplikasi tidak perlu membuat fungsi serupa atau menulis kode dari awal[3]. Penggunaan API AI dalam sistem kasir memungkinkan pengenalan dan pemberian deskripsi produk secara otomatis kepada kasir dan pelanggan, meningkatkan efisiensi dan pengalaman pengguna. Integrasi API juga memungkinkan aplikasi ini berkomunikasi dengan sistem lain, seperti platform pembayaran, manajemen inventaris, dan analitik penjualan, menciptakan ekosistem yang terhubung dan efisien. Hal ini membantu kedai kopi mengoptimalkan

operasional internal dengan menganalisis data transaksi dan preferensi pelanggan, memberikan rekomendasi produk yang relevan dan layanan personalisasi, serta mempermudah otomatisasi berbagai tugas operasional.

Untuk membuat aplikasi *mobile* diperlukan penggunaan flutter. Flutter adalah SDK yang dikembangkan oleh Google untuk membangun aplikasi *mobile* yang memiliki kinerja tinggi dan dapat berjalan pada platform iOS dan Android dengan menggunakan satu basis kode. SDK ini bersifat open source, sehingga dapat diakses dan dimodifikasi oleh siapa saja. Tujuan utama dari Flutter adalah memungkinkan pengembang untuk menciptakan aplikasi yang memiliki kinerja tinggi dan terasa alami pada berbagai platform[4]. Dalam pembuatannya pengembang perlu memahami bahasa pemrograman Dart. Dart adalah bahasa pemrograman yang juga dikembangkan oleh Google dan dimaksudkan sebagai pengganti Javascript.

Dengan demikian, sistem kasir berbentuk *mobile* dengan integrasi API AI telah menjadi pendorong utama inovasi dalam industri *coffee shop* modern[5]. Dengan memanfaatkan teknologi ini, *coffee shop* dapat meningkatkan efisiensi operasional mereka, memberikan pengalaman pelanggan yang lebih baik, dan tetap relevan dalam persaingan pasar yang semakin ketat[6].

2. TINJAUAN PUSTAKA

2.1. Flutter

Flutter adalah kerangka kerja open-source yang dikembangkan oleh Google untuk membangun antarmuka pengguna (UI) aplikasi secara cepat dan konsisten di berbagai platform, termasuk Android, IOS, dan web. Dengan menggunakan bahasa pemrograman Dart, Flutter memungkinkan pengembang untuk membuat aplikasi dengan antarmuka yang kaya dan responsif, serta performa yang tinggi. Flutter juga menyediakan berbagai komponen dan *widget* yang dapat digunakan untuk mempercepat proses pengembangan aplikasi, sehingga memberikan pengalaman pengembangan yang efisien dan memudahkan pengembang untuk menciptakan aplikasi *mobile* yang berkualitas tinggi[7].

2.2. Aplikasi Mobile

Aplikasi *mobile* adalah perangkat lunak yang dirancang khusus untuk digunakan pada perangkat seluler, seperti *smartphone* dan tablet. Dalam beberapa tahun terakhir, aplikasi *mobile* telah menjadi teknologi yang paling populer dan banyak digunakan[8].

Hal ini terutama didorong oleh perkembangan pesat dalam teknologi *smartphone* yang semakin canggih, yang menyediakan platform yang kuat untuk menjalankan berbagai jenis aplikasi dengan berbagai fungsi dan tujuan. Aplikasi *mobile* memungkinkan pengguna untuk melakukan berbagai aktivitas, mulai dari komunikasi, hiburan, belanja online, hingga

produktivitas, dengan mudah dan nyaman melalui perangkat seluler mereka.

2.3. Dart

Dart adalah bahasa pemrograman yang telah dikembangkan oleh Google sejak tahun 2007 di bawah kepemimpinan Lars Bak dan Kasper Lund[9].

Dart merupakan bahasa pemrograman yang berorientasi objek (OOP) dengan syntax yang mirip dengan C++, Java dan Javascript. Bahasa ini dirancang untuk mengoptimalkan sisi klien. Selain untuk pengembangan aplikasi seluler, Dart juga dapat digunakan untuk membuat berbagai jenis aplikasi, termasuk aplikasi web, layanan mikro, desktop, dan aplikasi lain yang menggunakan teknologi *Internet of Things* (IoT).

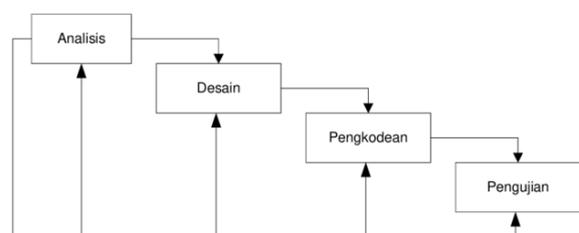
2.4. API Gemini

Gemini adalah model *Large Language Model* (LLM) dari Google yang sangat besar, canggih, dan fleksibel. Model ini dapat dijalankan di berbagai mesin, mulai dari data center hingga perangkat *mobile*, dan mampu memahami berbagai jenis informasi seperti teks, kode, suara, gambar, dan video. Google membuka akses ke kunci API Gemini agar pengembang aplikasi dapat menggunakan model ini dalam pembuatan aplikasi kecerdasan buatan (AI) mereka sendiri[10].

Terdapat contoh penelitian sebelumnya yang menerapkan Google API telah berhasil membangun aplikasi Tour Guide yang dapat menampilkan lokasi keberadaan user, informasi tentang objek wisata, rute perjalanan, jarak terdekat maupun terjauh serta waktu tempuhnya dengan pilihan navigasi[11].

3. METODE PENELITIAN

Pengembangan adalah sebuah proses yang bertujuan untuk meningkatkan atau menciptakan sesuatu yang baru. Metode yang digunakan untuk pengembangan sistem kasir ini adalah metode *waterfall* yang telah disederhanakan. Model *waterfall* memberikan pendekatan berurutan terhadap alur hidup perangkat lunak. Berikut ini tahapan penelitian ini, yaitu:



Gambar 1. Diagram Alur Penelitian

3.1. Analisis

Tahap analisis kebutuhan sistem dalam pengembangan aplikasi sistem kasir *coffee shop* berbasis *mobile* sangat penting untuk memahami kebutuhan pengguna serta mengidentifikasi komponen

yang diperlukan. Proses ini juga membantu dalam memahami manfaat yang akan diberikan oleh aplikasi yang dikembangkan. Analisis ini mencakup identifikasi perangkat keras, perangkat lunak, serta kebutuhan input, proses, dan output aplikasi. Dalam konteks pengembangan aplikasi berbasis *mobile*, analisis kebutuhan juga mencakup penilaian kemampuan portabilitas, kemudahan penggunaan, dan kemampuan *offline*[12].

Sebuah aplikasi sistem kasir *coffee shop* yang telah ada mempunyai tiga halaman utama dengan fitur seperti *view* detail produk, tambah produk, edit produk, hapus produk, tambah pesanan, *view* keranjang dan *view* laporan. Permasalahan ditemukan yaitu sistem memerlukan fitur yang dapat menghemat waktu dan memberikan kemudahan kepada pengguna untuk mendeskripsikan berbagai macam produk. Ini menyebabkan terciptanya sistem kasir dengan fitur *generate* deskripsi untuk membuat deskripsi produk otomatis tanpa membuat deskripsi manual dan memasukkannya ke dalam *database*.

3.2. Desain

Desain bertujuan untuk merancang arsitektur sistem kasir yang memenuhi kebutuhan dan persyaratan yang telah diidentifikasi sebelumnya. Berdasarkan analisis sebelumnya, penerapan kecerdasan buatan dipertimbangkan. Dengan menggunakan model AI Gemini untuk membuat aplikasi kasir *mobile* dengan fungsionalitas pembuatan deskripsi produk untuk membantu pengguna menghemat waktu menulis deskripsi produk.

Untuk memenuhi kebutuhan yang telah dianalisis sebelumnya, perancangan arsitektur sistem kasir dilakukan. Dengan gambaran rancangan, pengguna dapat mengakses (*view*) detail produk kemudian dapat melakukan operasi *generate* deskripsi dari produk tertentu yang dipilih dalam aplikasi kasir *mobile*. Fokus utama pada tahap ini adalah pengembangan tata letak antarmuka untuk memastikan pengalaman pengguna yang optimal di perangkat *mobile*[13].

3.3. Pengkodean

Dalam tahap pengkodean, peneliti menggunakan berbagai teknologi dan alat seperti Flutter, bahasa Dart, *query*, variabel, kecerdasan buatan (AI), dan komponen pendukung lainnya untuk mengembangkan aplikasi sistem kasir *mobile* ini. Implementasi API AI Gemini yang telah selesai dikembangkan dan melewati proses pengujian. Integrasi API dengan sistem kasir berbasis *mobile* membutuhkan perencanaan dan pengujian yang matang untuk memastikan interoperabilitas yang baik antara API dan aplikasi. Setelah implementasi, aplikasi siap digunakan oleh pengguna, yaitu para kasir *coffee shop*. Tujuan dari tahap ini adalah untuk menghasilkan aplikasi yang sesuai dengan tahap desain sistem yang telah dirancang sebelumnya[14].

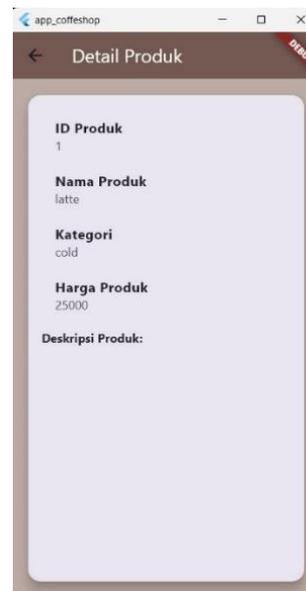
3.4. Pengujian

Pengujian sistem dilakukan untuk memverifikasi dan memvalidasi aplikasi kasir yang dikembangkan. Pengujian mencakup pengujian fungsional dan non-fungsional, seperti pengujian integrasi antarmuka pengguna, keamanan, dan kinerja. Dalam konteks pengembangan aplikasi berbasis *mobile*, peneliti menggunakan metode blackbox testing adalah teknik pengujian software yang memeriksa hasil keluaran berdasarkan nilai masukan tanpa memerlukan perhatian terhadap detail internal sistem tersebut[15].

4. HASIL DAN PEMBAHASAN

4.1. Desain

Pengguna akan masuk ke dalam halaman aplikasi *mobile* yang menampilkan list produk. Setelah itu pengguna dapat melakukan aktivitas *generate* dengan menekan salah satu produk dan muncul deskripsi produk otomatis dalam waktu singkat. Setelah selesai pengguna dapat kembali ke halaman sebelumnya dan melakukan *generate* deskripsi produk yang lainnya dengan menekan produk lainnya.



Gambar 2. Tampilan Detail Produk Sebelum Inisiasi API Gemini

4.2. Pengkodean

Tahap pengkodean yang diperlukan adalah dimulai dengan meng-*install* flutter_gemini sebagai *library* dengan flutter *command* ‘flutter pub add flutter_gemini’, serta menambahkan versi flutter_gemini ke *package* pubspec.yaml dan menjalankan ‘flutter pub get’:

```
dependencies:
  flutter_gemini: ^2.0.3
```

Setelah selesai meng-*install* gemini pada flutter maka *library* flutter_gemini dapat di *import* pada halaman main.dart untuk menginisiasi api key Gemini pada *void main()* seperti berikut:

```
void main() {
  runApp(const MyApp());
  // Inisialisasi objek Gemini saat aplikasi dimulai
  Gemini.init(apiKey: 'XXAPI_KEYXX');
}
```

Kemudian, *library flutter_gemini* juga di import pada halaman *detail_produk.dart* untuk menginisiasikan gemini pada aplikasi *mobile* kasir dengan membuat fungsi yang nantinya dapat dipanggil ketika kasir menekan salah satu produk, sebagai berikut:

```
void initState() {
  super.initState();
  // Mendapatkan objek Gemini yang sudah diinisialisasi
  gemini = Gemini.instance;

  // Panggil fungsi untuk menghasilkan deskripsi produk saat layar pertama kali dibuka
  _generateDesc();
}

Future<void> _generateDesc() async {
  try {
    // Memanggil model AI untuk menghasilkan deskripsi produk
    final response =
      await gemini.text("Buat Deskripsi $productName Sebanyak Minimal 15 Kata");

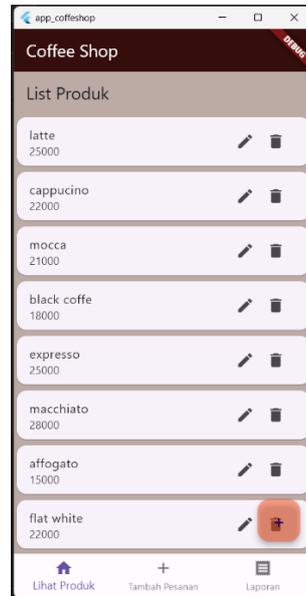
    // Mengupdate state untuk menampilkan deskripsi produk
    setState() {
      storyText = response?.output;
    });
  } catch (e) {
    print(e);
  }
}
```

Dengan pembuatan fungsi seperti diatas, model AI yang telah dibuat memungkinkan aplikasi untuk menghasilkan deskripsi produk minimal 15 kata pada halaman produk detail, lalu menampilkannya kepada kasir setelah menekan nama salah satu produk pada halaman *list* produk *coffee shop*.

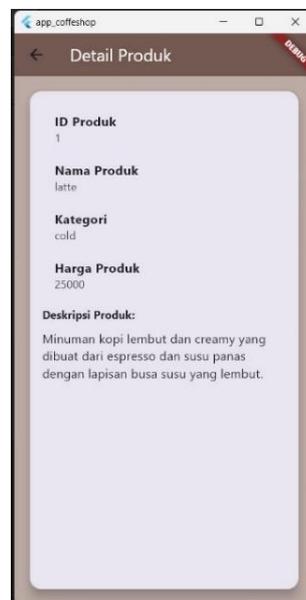
4.3. Pengujian

Setelah tahap pengkodean fungsi *generate* dilakukan, fokus utama selanjutnya adalah pengujian

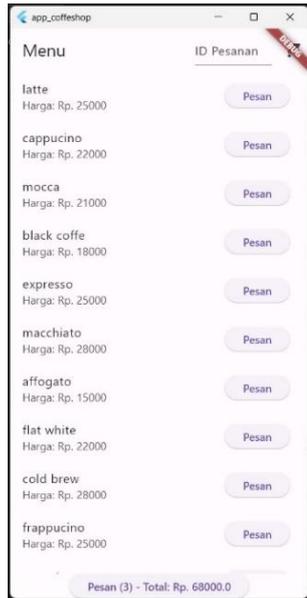
pada fungsionalitas fungsi *generate* yang memicu API untuk menghasilkan deskripsi yang sesuai dengan permintaan pengguna terkait produk. Alur pengujian dimulai dengan memastikan bahwa fungsi *generate* berfungsi dengan baik dan mengirimkan permintaan yang tepat ke API. Setelah produk dipilih, API diuji untuk memastikan bahwa output deskripsi yang dihasilkan sesuai dengan harapan dengan menggambarkan barang secara akurat dan lengkap.



Gambar 3. Halaman Produk



Gambar 4. Tampilan Hasil Fungsi *Generate*



Gambar 5. Halaman Tambah Pesanan



Gambar 6. Halaman Laporan

Tabel 1. Black Box Testing

No	Deskripsi pengujian	Hasil yang diharapkan	Keterangan
1	Halaman CRUD Produk	Dapat menambah data produk, menghapus data produk, mengedit data produk dan Menampilkan detail data produk.	Sesuai
2	Generate Deskripsi Produk Saat Laman Detail Produk Dibuka	Fungsi <i>generate</i> berhasil membaca nama produk dan berhasil secara otomatis mengeluarkan output deskripsi produk minimal 15 kata	Sesuai
3	Halaman Tambah Pesanan	Dapat menambahkan pesanan kedalam keranjang dan menghitung total harga pesanan	Sesuai
4	Halaman Laporan	Dapat menampilkan seluruh laporan pemesanan yang telah dibuat.	Sesuai

Hasil pengujian menunjukkan bahwa API Gemini *Generate* Deskripsi berfungsi dengan baik pada percobaan dan menghasilkan deskripsi yang sesuai dengan harapan hanya dalam hitungan detik (sekitar 2-5 detik), tanpa masalah signifikan yang ditemukan dalam proses *generate* deskripsi.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengembangan yang dilakukan, dapat disimpulkan bahwa pengembangan sistem kasir untuk *coffee shop* berbasis *mobile*, dengan tahapan pengembangan meliputi analisi, desain, pengkodean, dan pengujian, memberikan efektivitas waktu kepada pengguna. Dalam proses pengembangan, implementasi AI Gemini digunakan untuk membantu pengguna membuat deskripsi produk hanya dalam aplikasi *mobile* tanpa perlu membuka aplikasi pendukung lainnya. Dengan pelaksanaan fase-fase pengembangan yang dilakukan secara bertahap, metode *Waterfall* yang disederhanakan dapat meminimalisir terjadinya kesalahan.

Pengembangan sistem kasir *coffee shop* berbasis *mobile* dengan fitur *generate* deskripsi produk berhasil memberikan solusi efektif untuk menghemat waktu dan memudahkan pengguna dalam mendeskripsikan berbagai macam produk. Melalui tahapan analisis kebutuhan, desain sistem, pengkodean, dan pengujian,

sistem kasir ini mampu menghasilkan deskripsi produk secara otomatis dengan menggunakan AI Gemini, yang mengurangi kebutuhan deskripsi manual dan memasukkannya ke dalam *database*. Pengujian menunjukkan bahwa fungsi *generate* deskripsi berfungsi dengan baik, menghasilkan deskripsi yang sesuai dengan harapan dalam waktu singkat (sekitar 2-5 detik), tanpa masalah signifikan yang ditemukan. Implementasi ini membuktikan bahwa integrasi teknologi AI dalam aplikasi *mobile* kasir dapat meningkatkan efisiensi dan kualitas layanan di *coffee shop*.

Saran yang dapat diberikan adalah untuk terus mengembangkan fitur-fitur baru dalam aplikasi dan meningkatkan fungsionalitas yang ada berdasarkan hasil pengujian aplikasi. Selain itu, penting untuk terus melakukan optimasi kinerja aplikasi guna memastikan bahwa aplikasi dapat berjalan dengan lancar dan efisien. Evaluasi internal secara berkala terhadap kode dan sistem juga diperlukan untuk menjaga kualitas dan keandalan aplikasi tersebut.

DAFTAR PUSTAKA

[1] D. Damayanti, "Rancang Bangun Sistem Pengukuran Keselarasan Teknologi Dan Bisnis Untuk Proses Auditing," *J. Tekno Kompak*, vol. 14, no. 2, p. 92, 2020, doi:

- 10.33365/jtk.v14i2.727.
- [2] I. G. M. Ariantara, I. Arwani, and W. H. N. Putra, "Penerapan REST API dalam Pengembangan Aplikasi Pemesanan Rental Mobil berbasis Web dan Mobile (Studi Kasus: CV. Dwi Cipta Rent Car)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 8, pp. 2569–2576, 2020.
- [3] A. Najib and M. Y. Zain, "Aplikasi Point of Sale Multi Outlet Dan Multi Payment Berbasis Web Dan Android," *Konvergensi*, vol. 16, no. 2, p. 55, 2020, doi: 10.30996/konv.v16i2.4102.
- [4] S. Tjandra and G. S. Chandra, "Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang," *J. Inf. Syst. Hosp. Technol.*, vol. 2, no. 02, pp. 76–81, 2020, doi: 10.37823/insight.v2i02.109.
- [5] G. Santoso, J. Setiawan, and A. Sulaiman, "Development of OpenAI API Based Chatbot to Improve User Interaction on the JBMS Website," *G-Tech J. Teknol. Terap.*, vol. 7, no. 4, pp. 1606–1615, 2023, doi: 10.33379/gtech.v7i4.3301.
- [6] Eka Yuniar and Heri Purnomo, "Implementasi Chatbot 'Alitta' Asisten Virtual Dari Balittas Sebagai Pusat Informasi Di Balittas," *Antivirus J. Ilm. Tek. Inform.*, vol. 13, no. 1, pp. 24–35, 2019, doi: 10.35457/antivirus.v13i1.714.
- [7] N. Erzed, N. Anwar, A. M. Widodo, E. Prasetyo, and K. K. Juman, "Implementasi Flutter Pada Aplikasi Presensi Karyawan Berbasis Mobile," *Ikraith-Informatika*, vol. 6, no. 3, pp. 100–106, 2022, doi: 10.37817/ikraith-informatika.v6i3.2211.
- [8] I. Larasati, A. N. Yusril, and P. Al Zukri, "Systematic Literature Review Analisis Metode Agile Dalam Pengembangan Aplikasi Mobile," *Sistemasi*, vol. 10, no. 2, p. 369, 2021, doi: 10.32520/stmsi.v10i2.1237.
- [9] K. Tampubolon, "Copyright@2023, ORAHUA, Hlm | 24 Pelatihan Pengenalan Dasar Bahasa Dart Dalam Pemograman OOP," *ORAHUA J. Pengabd. Kpd. Masy.*, vol. 1, no. 1, pp. 24–29, 2023.
- [10] N. Rachmat and D. P. Kesuma, "Implementasi LLM Gemini Pada Pengembangan Aplikasi Chatbot Berbasis Android," *J. Ilmu Komput.*, vol. 4, no. 1, pp. 40–52, 2024.
- [11] D. A. Purnamasari, V. Karisa, and A. Uperiati, "Rancang Bangun Aplikasi Pemandu Wisata di Kota Batam dengan Penerapan Google API & PWA," vol. 10, no. 2, pp. 95–101, 2023.
- [12] M. Kulia, S. Asean, A. R. Asmi, A. Novemy, D. Surbakti, and Y. Hasan, "Pengembangan Model Mobile Learning Berbasis Aplikasi Android pada," vol. 5, no. 1, pp. 30–40, 2019.
- [13] Nova Noor Kamala Sari, Putu Bagus Adidyana Anugrah Putra, and Efrans Christian, "Rancang Bangun Aplikasi Mobile Learning Tenses Bahasa Inggris," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 13, no. 2, pp. 37–46, 2019, doi: 10.47111/jti.v13i2.253.
- [14] N. H. Adi, A. R. Basar, S. Umar, and A. T. Devega, "Virtual Asisten Pusat Informasi Mahasiswa Berbasis Android Menggunakan Flutter," vol. 1, no. 2, 2022, doi: 10.55585/rintvet.v1i2.16.
- [15] A. A. Arwaz, T. Kusumawijaya, R. Putra, K. Putra, and A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 130, 2019, doi: 10.32493/jtsi.v2i4.3708.